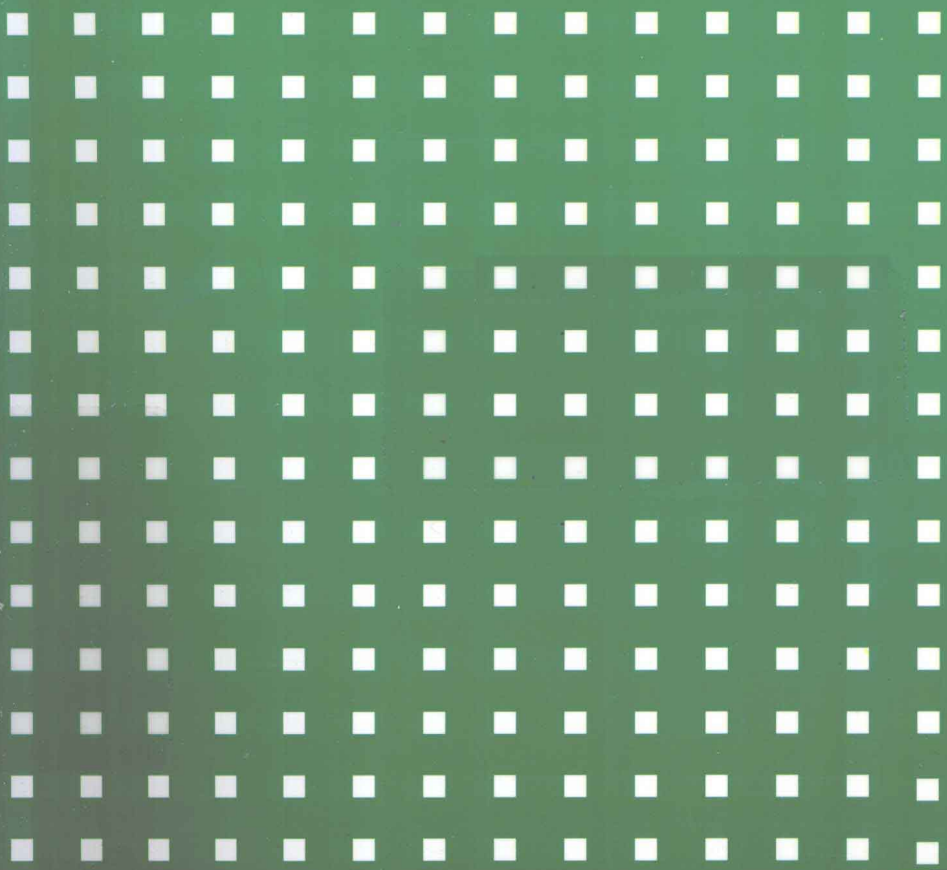


高等学校计算机专业教材精选·算法与程序设计

# Java项目实战教程

姜 华 刘 闯 主编

邱秀伟 孙静怡 赵洪强 副主编



清华大学出版社

高等学校计算机专业教材精选·算法与程序设计

# Java项目实战教程

姜 华 刘 闯 主 编  
邱秀伟 孙静怡 赵洪强 副主编

清华大学出版社  
北京

## 内 容 简 介

本书以培养应用型人才为目标,以项目案例开发为主线,全面而又系统地介绍 Java 编程技术。全书共分 12 章,内容包括 Java 开发基础、Java 语法基础、类和对象、继承和多态、接口和常量、集合框架、异常处理、数据库连接、JSP 开发基础、JSP 实现数据交互、JSP 优化处理和 JavaEE 框架。

本书既可作为高校计算机及其相关专业教材,也可供与计算机相关专业的技术人员使用,尤其适合有一定面向对象编程基础的数据库应用开发、JSP 的 Web 应用开发人员阅读参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

Java 项目实战教程/姜华,刘闯主编.--北京:清华大学出版社,2012.9

高等学校计算机专业教材精选·算法与程序设计

ISBN 978-7-302-29393-4

I. ①J… II. ①姜… ②刘… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 159737 号

责任编辑:白立军 顾 冰

封面设计:傅瑞学

责任校对:时翠兰

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京市人民文学印刷厂

装 订 者:三河市兴旺装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:17.5

字 数:425 千字

版 次:2012 年 9 月第 1 版

印 次:2012 年 9 月第 1 次印刷

印 数:1~3000

定 价:29.50 元

---

产品编号:046910-01

# 前 言

Java 语言作为一种优秀的面向对象的程序设计语言,具有平台无关性、安全机制、高可靠性和内嵌的网络支持等特点,是当前网络应用程序编写的首选工具。

Java 程序设计是一门实践性、专业性、实用性和可操作性很强的课程,要求在掌握基本操作理论和技能的基础上,注重项目开发能力的培养,以项目驱动的案例教学方法效果较佳。但在目前实际的教学中,针对本科生以项目案例开发为主的应用型教材不多,导致学生所用教材和教学目标脱节。因此,编者结合长期的一线教学实践,编写了这样一本面向应用型本科院校计算机及其相关专业,以项目驱动、案例实战为主的 Java 教材。

本书按照应用型人才培养的教学特点,从培养 Java 项目开发能力、注重 Java 项目开发技术的实用出发,以 JavaEE 主流框架整合应用及项目开发为主线,通过 Java Web 开发中最常见的典型模块和项目案例,全面地介绍了 Java 基本设计、Struts 2. x、JSP 等应用技术及 Java 项目开发的过程。在内容安排上,贯穿由浅入深、循序渐进的原则,符合认知规律。

全书共 12 章。第 1~7 章介绍 Java 编程基本知识,包括基本语法结构、面向对象编程的核心内容及 Java 集合框架等;第 8~12 章介绍 JavaEE 平台开发技术,包括数据库的 JDBC API、Servlets、JSP 开发、Struts/Hibernate/Spring 基本原理等。在注重系统性和科学性的同时,力求突出实用性,在介绍相关编程原理和基础知识的同时,着重利用丰富的实例演示。每章均附有一定数量的案例,分成上课案例和练习案例,在技术介绍过程中采用的上课案例,突出相关技术核心内容,帮助读者掌握使用 Java 进行系统分析、设计和实现的基本思路和方法。练习案例是结合对课堂知识的消化吸收,帮助读者完成项目的设计开发,提高编程能力。

本书第 1~3 章由姜华编写,第 4~6 章由邱秀伟编写,第 7 章由孙静怡编写,第 8~12 章由刘闯编写,全书由姜华统稿。

本书在编写过程中得到许多同行、专家及领导的关心和支持,在此表示衷心感谢。

虽然笔者非常认真地编写、校对本书全部内容,但由于时间仓促以及作者水平有限,错误和不足之处在所难免,恳请读者批评指正。

编 者  
2012 年 4 月

# 目 录

<b>第 1 章 Java 开发基础</b> .....	<b>1</b>
1.1 Java 的开发环境简介 .....	1
1.1.1 Java 软件开发工具箱 .....	1
1.1.2 Java 集成开发环境 .....	1
1.2 Java 程序结构 .....	2
1.3 Java 注解及内嵌式文档 .....	5
1.3.1 Java 程序注解 .....	5
1.3.2 Java 的内嵌式文档 .....	5
1.3.3 Java 的编码风格 .....	6
1.4 Java 帮助文档 .....	7
1.5 项目练习 .....	7
1.5.1 项目概述 .....	7
1.5.2 上机任务 1 .....	7
1.5.3 上机任务 2 .....	9
<b>第 2 章 Java 语法基础</b> .....	<b>10</b>
2.1 Java 运算符 .....	10
2.1.1 赋值运算符 .....	10
2.1.2 算术运算符 .....	11
2.1.3 关系运算符 .....	11
2.1.4 特殊运算符 .....	12
2.2 数据类型和变量 .....	13
2.2.1 数据类型 .....	13
2.2.2 变量 .....	14
2.3 字符串 .....	14
2.3.1 创建字符串 .....	14
2.3.2 字符串基本操作 .....	14
2.4 流程控制 .....	18
2.4.1 条件选择语句 .....	18
2.4.2 循环语句 .....	20
2.4.3 控制语句 .....	21
2.5 数组 .....	22
2.5.1 Java 中的数组 .....	22
2.5.2 数组的应用 .....	24

2.6	项目练习	26
2.6.1	上机任务 1	26
2.6.2	上机任务 2	27
2.6.3	上机任务 3	29
<b>第 3 章</b>	<b>类和对象</b>	<b>32</b>
3.1	类	32
3.1.1	类和对象的区别	32
3.1.2	类的定义	32
3.2	对象	34
3.2.1	封装	34
3.2.2	对象的创建和使用	34
3.3	类的方法	36
3.3.1	类方法的定义	36
3.3.2	类方法的调用	36
3.4	包	38
3.4.1	包的创建	38
3.4.2	包和类的导入	39
3.5	封装	40
3.6	构造方法	42
3.7	项目练习	45
3.7.1	项目概述	45
3.7.2	上机任务 1	45
3.7.3	上机任务 2	47
3.7.4	上机任务 3	47
3.7.5	上机任务 4	51
<b>第 4 章</b>	<b>继承和多态</b>	<b>54</b>
4.1	继承	54
4.1.1	继承的概念	54
4.1.2	继承的实现	57
4.1.3	super 关键字	57
4.2	多态	58
4.2.1	重写和重载	58
4.2.2	this 关键字	63
4.3	项目练习	63
4.3.1	上机任务 1	63
4.3.2	上机任务 2	65

<b>第 5 章 接口和常量</b>	<b>68</b>
5.1 接口	68
5.1.1 Java 接口	68
5.1.2 扩充接口	71
5.2 常量	72
5.2.1 常量	72
5.2.2 接口中的常量	73
5.3 项目练习	74
5.3.1 上机任务 1	74
5.3.2 上机任务 2	75
5.3.3 上机任务 3	77
<b>第 6 章 集合框架</b>	<b>79</b>
6.1 集合框架	79
6.1.1 集合框架简介	79
6.1.2 集合框架体系	79
6.2 List 接口及实现	81
6.2.1 ArrayList 类	81
6.2.2 LinkedList 类	85
6.2.3 Iterator 接口	86
6.3 Set 接口及实现	87
6.3.1 Set 接口	87
6.3.2 HashSet 类	88
6.3.3 TreeSet 类	89
6.4 Map 接口及其实现	92
6.4.1 HashMap 用法	92
6.4.2 TreeMap 用法	94
6.5 泛型集合	95
6.6 项目练习	97
6.6.1 上机任务 1	97
6.6.2 上机任务 2	99
6.6.3 上机任务 3	99
6.6.4 上机任务 4	100
6.6.5 上机任务 5	100
<b>第 7 章 异常处理</b>	<b>102</b>
7.1 Java 异常	102
7.1.1 异常简介	102
7.1.2 异常的产生	102

7.1.3	Java 的异常类 .....	103
7.2	异常处理机制 .....	104
7.2.1	捕捉异常 .....	104
7.2.2	finally .....	108
7.2.3	抛出异常 .....	109
7.2.4	声明异常 .....	110
7.2.5	编写自己异常类 .....	111
7.3	异常日志 .....	113
7.3.1	log4j .....	113
7.3.2	log4j 使用 .....	114
7.4	项目练习 .....	119
7.4.1	上机任务 1 .....	119
7.4.2	上机任务 2 .....	120
<b>第 8 章</b>	<b>数据库连接 .....</b>	<b>122</b>
8.1	JDBC 简介 .....	122
8.1.1	什么是 JDBC .....	122
8.1.2	JDBC 的功能 .....	122
8.1.3	JDBC 与 ODBC .....	123
8.2	JDBC 工作原理 .....	124
8.2.1	JDBC 操作步骤 .....	124
8.2.2	JDBC 结构 .....	125
8.2.3	JDBC URL .....	125
8.3	JDBC 驱动 .....	126
8.3.1	JDBC 常见驱动 .....	126
8.3.2	常用数据库的 JDBC 驱动 .....	127
8.4	通过 JDBC 访问数据库 .....	128
8.4.1	访问数据库步骤 .....	128
8.4.2	利用 Statement 语句操作数据库 .....	133
8.4.3	利用 PreparedStatement 语句操作数据库 .....	137
8.4.4	批量处理 JDBC 语句 .....	142
8.5	项目练习 .....	144
8.5.1	上机任务 1 .....	144
8.5.2	上机任务 2 .....	147
8.5.3	上机任务 3 .....	148
<b>第 9 章</b>	<b>JSP 开发基础 .....</b>	<b>149</b>
9.1	JSP 开发环境配置 .....	149
9.1.1	JSP 与 B/S 技术 .....	149



9.1.2	开发环境配置	150
9.2	JSP 开发 Web 项目步骤	152
9.2.1	创建 Web 项目	153
9.2.2	编写代码	153
9.2.3	部署并运行 Web 项目	154
9.3	JSP 简介	156
9.3.1	JSP 页面基本结构	157
9.3.2	编写 JSP 页面	160
9.3.3	设置 JSP 的欢迎页	160
9.4	JSP 语法	161
9.4.1	JSP 指令	161
9.4.2	服务器端脚本元素	165
9.5	项目练习	167
9.5.1	上机任务 1	167
9.5.2	上机任务 2	170
<b>第 10 章</b>	<b>用 JSP 实现数据交互</b>	<b>175</b>
10.1	处理客户端请求	175
10.1.1	表单数据获取	175
10.1.2	JSP 内置对象	177
10.1.3	转发与重定向	184
10.2	JSP 访问数据库技术	187
10.2.1	JSP 访问数据库	187
10.2.2	数据库连接池技术	192
10.2.3	JSP 分页技术	195
10.3	JSP 实现访问控制	201
10.3.1	session 对象	201
10.3.2	application 对象	204
10.3.3	Cookie 对象	206
10.3.4	exception 对象	210
10.4	项目练习	211
10.4.1	上机任务 1	211
10.4.2	上机任务 2	212
10.4.3	上机任务 3	213
10.4.4	上机任务 4	213
<b>第 11 章</b>	<b>JSP 优化处理</b>	<b>215</b>
11.1	JavaBean 技术	215
11.1.1	JavaBean 简介	215

11.1.2	JavaBean 在 JSP 中的使用 .....	217
11.2	JSP 常用标签 .....	221
11.2.1	<jsp:forward> 标签 .....	221
11.2.2	<jsp:include> 标签 .....	222
11.2.3	<jsp:param> 标签 .....	222
11.3	EL 表达式 .....	226
11.3.1	EL 基本语法 .....	227
11.3.2	EL 常用隐含对象 .....	227
11.4	项目练习 .....	230
11.4.1	上机任务 1(可选) .....	230
11.4.2	上机任务 2(可选) .....	231
11.4.3	上机任务 3(可选) .....	234
<b>第 12 章</b>	<b>JavaEE 框架 .....</b>	<b>237</b>
12.1	Struts 框架基本原理 .....	237
12.1.1	Struts 简介 .....	237
12.1.2	Struts 简单应用 .....	238
12.2	Spring 框架基本原理 .....	252
12.2.1	Spring 简介 .....	252
12.2.2	Spring 简单应用 .....	253
12.3	Hibernate 基本原理 .....	256
12.3.1	Hibernate 简介 .....	256
12.3.2	Hibernate 简单应用 .....	257
12.4	Struts、Spring 和 Hibernate 的集成 .....	264
12.4.1	Spring 与 Hibernate 的整合 .....	265
12.4.2	Spring 与 Struts 的整合 .....	267
	<b>参考文献 .....</b>	<b>268</b>

# 第 1 章 Java 开发基础

## 本章要点

- Java 开发环境
- Java 程序结构
- Java 的注解和文档

Java 编程人员大部分是通过 Oracle 公司的 JDK (Java Developers Kits) 学习 Java 的, 通过这个免费的 JDK 开发工具包可以开发 Java 程序。随着应用的深入, 目前大部分的 Java 项目开发都通过集成化开发环境来实现。

## 1.1 Java 的开发环境简介

### 1.1.1 Java 软件开发工具箱

Java 作为一门高级程序语言, 需要经过编译、链接等才能运行。Oracle 公司提供的开发工具包 JDK 可以从 Oracle 的官方网站地址 <http://www.oracle.com/technetwork/java/javase/downloads/index.html> 免费下载。本书使用 JDK6 来开发程序。下载完工具包如: `jdk-6-windows-i586.exe`, 就可以直接安装。安装完成之后, 需要对环境变量做相关配置, 这样才能较好地执行 Java 命令。

在 Windows XP 中的环境变量配置如下:

(1) 在 Windows 桌面上右击“我的电脑”然后单击“属性”选项, 在“系统属性”对话框中选择“高级”选项卡, 然后单击“环境变量”按钮, 打开“环境变量”对话框, 找到“系统变量”中变量名为 Path 的行, 如图 1.1 所示。

(2) 单击“编辑”按钮, 将 JDK 中的 bin 目录的安装位置加入到环境变量中。具体做法是, 在当前变量值的最后加入“;”, 然后将安装 bin 的完整路径加入即可, 如图 1.2 所示。

### 1.1.2 Java 集成开发环境

当完成基本的 JDK 安装之后就利用记事本等文本编辑工具编写 Java 源程序。但是用此类单纯的字符编辑工具编写源程序效率比较低, 并且还要通过 JDK 工具进行编译、运行。现在大部分的 Java 开发都是用集成开发环境(开发工具), 常见的有 JBuilder、NetBeans、MyEclipse 等。此类集成开发工具包含了编辑、编译、运行、调试等多种工具。这里主要介绍使用 MyEclipse 来开发 Java 项目。从 MyEclipse 的官方网站 <http://www.myeclipseide.com> 可以免费下载。安装完成之后就可以进行 Java 项目的开发。在使用该软件开发之前需要对其做简单配置, 启动 MyEclipse, 依次选择 Window | Preferences | Java | Installed JREs 命令, 然后在该窗口的右侧选择 Add 按钮, 将之前安装的 JDK 包中的 JRE 添加并选中即可, 如图 1.3 所示。

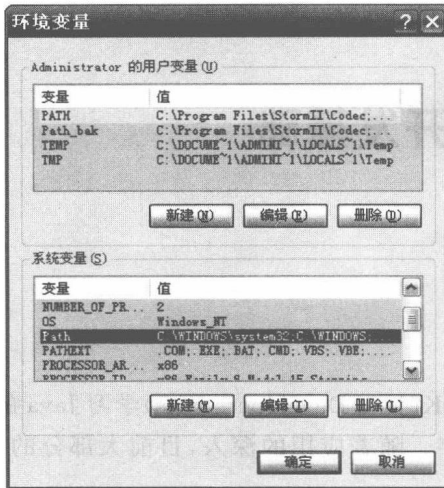


图 1.1 “环境变量”对话框



图 1.2 Path 环境变量配置

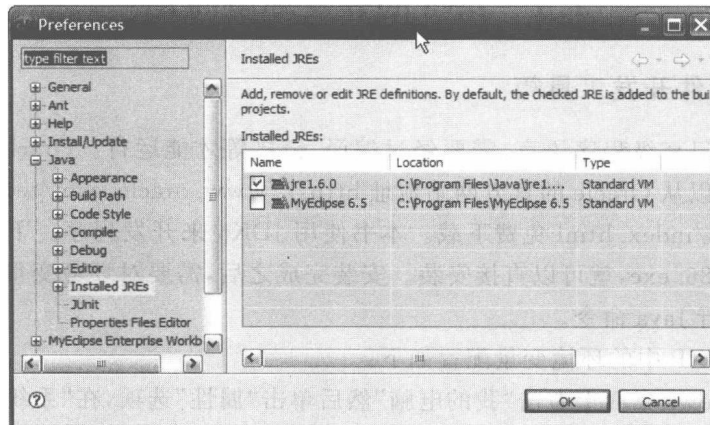


图 1.3 在 MyEclipse 中配置 JRE

在 MyEclipse 中开发 Java 程序需要下面几个步骤。

### 1. 创建 Java 项目文件

选择 File|New|Java Project 然后输入项目文件名称即可，项目文件即完成该项目的一组文件的集合。

### 2. 编写 Java 源程序

在当前窗口中编写源代码，最后保存为 .java 文件。

### 3. 编译及运行 Java 源程序

直接单击“运行”按钮，将编译和运行一次完成。

## 1.2 Java 程序结构

Java 程序有基本的框架，一般包含类名、main 框架、基本代码三部分。Java 程序的主文件名必须和类名一致，下面以 HelloJava.java 程序为例说明。

### 例 1.1 简单的 Java 应用程序。

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

程序输出结果如图 1.4 所示。

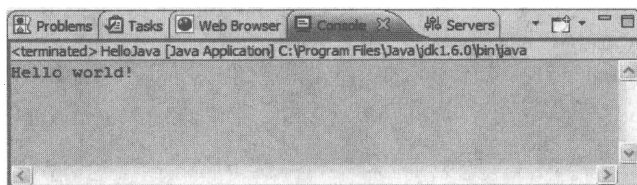


图 1.4 例 1.1 结果示意图

程序说明：

- ① class 的名称要与 Java 程序的主文件名称相同，否则编译过程会出错。
- ② 程序中必须有个“入口”，即 main 方法，一个程序最多只能有一个 main 方法，该标记必须是：

```
public static void main(String[] args) {}
```

其中 public 表示这是一个公开使用的函数。传入 main() 函数的参数是 String 对象数组。该程序中的 args 没有用到，但是不能省略，args 用来存储“命令行参数”。

③ System.out.println("Hello world!"); 为基本代码，System.out 表示标准输出，即显示器。调用标准输出流的 println() 方法向显示器输出字符串，该语句的作用是向显示器输出一条语句。

④ Java 语言编程区分大小写，Class 和 class 代表不同含义，只有后者才是声明类的修饰符。

下面介绍编译与运行。

(1) 利用 JDK 开发 Java 程序。

当安装好 JDK 并设置好路径之后即可编译和运行 Java 源程序。首先将当前目录转入 Java 源程序所在目录，然后在命令提示符下输入“javac 文件名.java”，如 javac HelloJava.java，如没有任何回应，表示编译成功。接着输入 java HelloJava，就会看到程序执行结果，如图 1.5 所示。其中，HelloJava.java 是源文件，Javac.java 是 Java 编译器。

(2) 利用 MyEclipse 开发 Java 程序。

安装好 MyEclipse 工具之后就可以来开发 Java 程序。首先创建一个 Java 项目，选择 File|New|Java Project 命令，在弹出的对话框中输入项目名称，单击 Finish 按钮即可完成项目创建。当完成项目创建之后就可以编写 Java 源程序，右击项目文件名选择 New|Class 命令，然后在弹出的对话框中输入类名，可以在下面复选框中选择是否包含 main 函数，如图 1.6 所示。

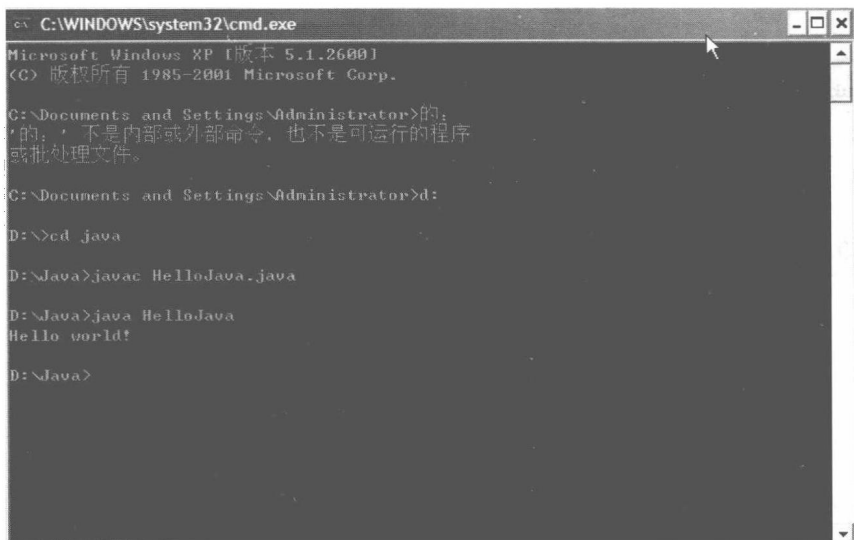


图 1.5 编译和运行 Java 源程序

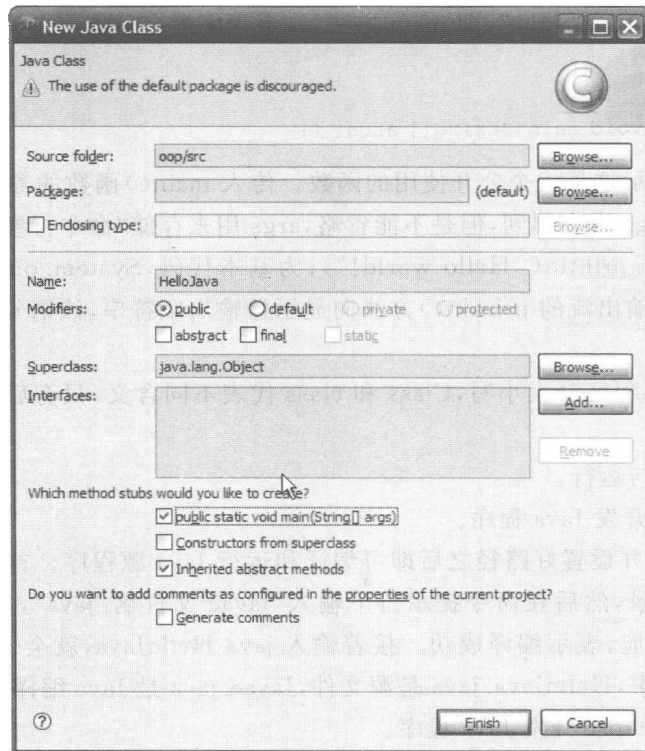



图 1.6 在项目创建 Java 程序

接下来就可以在窗口中编写程序代码了,代码编写完成之后就可以利用 MyEclipse 的工具来编译和运行,通过选择 Run|Run 命令(或在工具栏直接单击  按钮)即可完成编译和运行,如果程序正确则显示结果,否则会有错误信息提示。

## 1.3 Java 注解及内嵌式文档

### 1.3.1 Java 程序注解

为了便于 Java 程序的阅读和维护,需要有对应的注解说明,Java 提供了两种注解方式。一种是传统的注解风格,以 /\* 开始,以 \*/ 结尾,之间的部分都认为注解,在编写过程中部分开发者喜欢在注解中以 \* 开头。

```
/*
 * HelloJava.java
 * 我的第一个 Java 程序
 */
```

上述写法也可以用以下方式描述:

```
/* HelloJava.java
我的第一个 Java 程序 */
```

Java 的另一种注解方式为单行注解方式,以 // 开始,直到该行结束。这种注解方式更为简洁,所以被广泛使用。可以在行与行之间,或代码行的后面对代码做简短注解,例如:

```
//command on one line
```

### 1.3.2 Java 的内嵌式文档

程序在开发完成之后都附加有相应的说明文档。在程序改动时,对应的文档也要维护,单独维护文档工作量很大,Java 提供了一种简洁的文档维护方法,使得在程序修改完成之后相应的文档会自动更新,这种方法就是 Java 内嵌式文档,即将文档写入程序中,需要时即可通过 Javadoc 工具很轻松地将程序代码的内嵌式文档提取出来,生成相应的开发文档,和源程序保持一致。Javadoc 的命令语句以 /\*\* 开始,以 \*/ 结束,有内嵌式 HTML 或文档标签(doc tag)两种方式。文档标签是以 @ 开始的命令行,该符号要处于注解的最前面。文档内容一般有 class、variable、method,其中对 class 的注解要在 class 定义之前,对 variable 的注解恰好在变量定义之前,method 的注解在函数定义之前。

#### 1. 内嵌的 HTML

Javadoc 可以将 HTML 的控制命令加到 HTML 文档中,因而可以像其他网页一样使用 HTML 格式对相应的文本描述排列美化。

#### 2. 文档标签

文档标签常用的有 Class 文档标签和 Method 文档标签,其中 Class 文档标签有 @version,模块的版本信息;@author,开发者的信息;@since,程序代码使用的最早版本。Method 文档使用的标签有 @param,方法中的参数;@return,描述返回值的意义;@throws,描述抛出的异常信息。对例 1.1 加上注解后的程序描述如例 1.2 所示。

**例 1.2** 内嵌式文档和注解示例。

```
/**
```

```

* @author Administrator
* @version 2.0
* @since 1.0
* /
public class hello {
    /**
     *
     * @param args
     *         array of string arguments
     * @return No return value
     * @throws exception
     *         No exceptions thrown
     */
    public static void main(String[] args) {
        //TODO Auto-generated method stub
        System.out.println("hello java");
    }
}

```

在 MyEclipse 中输入/\*\*后自动显示注解的格式,并且输入@时会在下拉列表中显示 JavaDoc 的所有标签。

### 1.3.3 Java 的编码风格

为了增加程序的可读性,便于日常的开发和维护,Java 有自己的编码规范。

#### 1. 命名规范

类名首字母应该大写;属性(成员变量)、方法、对象变量以及所有标识符(如形式参数、实际参数、局部变量)的首字母应小写,其中包含的所有单词都应紧靠在一起,而且大写中间单词的首字母。例如:

- 类名: ThisIsAClassName;
- 属性或方法名: thisIsMethodOrFieldName;
- 对象变量: thisIsAClassVariable。

Java 包(Package)属于一种特殊情况,它们全都是小写字母,包括中间的单词。例如:

```
package hotlava.net.stats
```

#### 2. 代码书写规范

对于连接在一起,代码较长的程序,可采用分行显示,第二行一般在第一行的基础上缩进两个空格或一个 Tab。大括号{}在使用时,如果不在一行代码中,左括号{应与右括号}上下对齐,大括号里的首行代码,必须在下一行,并且缩进两个空格或一个 Tab。MyEclipse 提供了规范 Java 编码的方法,即采用按 Ctrl+Shift+F 键的方式实现对当前的程序文档的快速美化。

例 1.3 MyEclipse 的快速格式化 Java 编码示例。

```
public class HelloJava {
```



```
public static void main(String[] args) {
    //TODO Auto-generated method stub
    int anString;
    //.....
}
void changeColor(int newHe) {
    //.....
}
}
```

## 1.4 Java 帮助文档

在 Java 中提供了大量的类库,实现了相应的功能,程序开发人员只需要调用即可。开发人员可以通过查询帮助文档获得方法的详细说明,通过登录 Java 的官方网站: <http://docs.oracle.com/javase/6/docs/api/> 下载或者查看关于 API 的详细信息。

在 MyEclipse 中,也可以通过 Javadoc 窗口中看到方法的详细信息。例如若想查看 `println()` 方法的详细信息,就可以在编辑区单击 `println()` 方法名,在下面的 Javadoc 窗口就会显示详细信息,如图 1.7 所示。

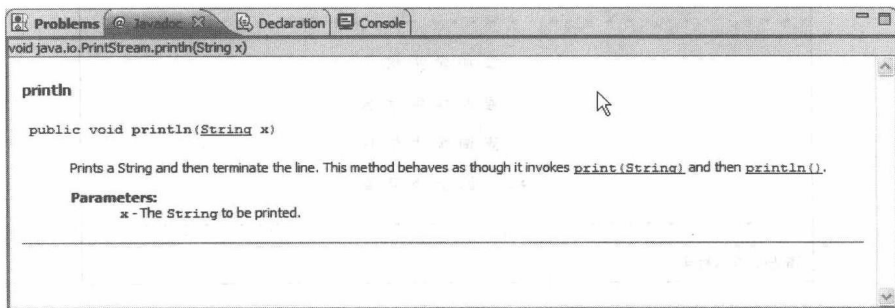


图 1.7 通过 Javadoc 窗口查看 `println()` 方法详细信息

## 1.5 项目练习

### 1.5.1 项目概述

本项目案例实现学生的基本信息管理。该信息管理系统包括:学生基本信息输入、学生信息查询、学生信息维护。要求在完成第 1 章和第 2 章的学习任务后完成系统的开发,其中第 1 章主要完成学生信息管理系统的菜单,第 2 章主要完成各个模块的详细功能。

### 1.5.2 上机任务 1

#### 1. 训练目标

- (1) 使用 MyEclipse 完成 Java 程序开发。
- (2) 熟悉 Java 基本输出语句。