

Java Web 应用开发 给力起飞



白灵 苟英 高博 周婕

编著

想知道Web程序是怎么从无到有的吗？

想了解商用Web程序是怎么构建自己的模块吗？

知或者不知，本书就在这里，不增不减，循序渐进。

你即将有机会坐上Web程序设计大巴，从基础走到新技术，从框架模块走到项目实现。沿途你将欣赏到社区系统实现的全部过程，详细了解Java Web编程的各种技术及其在实际开发中的应用。



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

http://www.phei.com.cn

Java Web

应用开发

给力起飞

白灵 苟英 高博 周婕

编著

电子工业出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书以项目开发任务为驱动,结合实际案例,讲解 Java Web 编程的各种技术及其在实际开发中的应用,包括 Java 面向对象编程基础,Java 数据库编程基础,JSP 编程基础,Hibernate、Struts 和 Spring 框架的基础及整合应用,Ajax 开发技术。在技术应用讲解中,一并介绍 Eclipse、MySQL 数据库及 Tomcat 服务器等软件的使用技巧。

全书以一个实用的 Web 系统——社区系统的开发为主线,系统的实现从简单到复杂,最终成为一个融合了 Ajax 技术的、标准 SSH2 (Struts2 + Spring + Hibernate) 架构的 Web 应用。

本书注重实用,知识精炼,讲解循序渐进、由浅入深,结合了丰富的案例和项目经验。对于有一定 Java 语言基础、想涉足 Java EE 领域的编程人员和爱好者,本书是不可多得的入门指引和参考用书;对于了解了一些 Java EE 相关知识,但对其在一个系统开发流程中的应用缺乏经验和体会的学习者,本书任务驱动的案例式讲解提供了手把手的帮助;而 Java 语言的初学者,也可以通过本书了解到 Java 语言的各种强大特性及其在实际开发中的作用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Java Web 应用开发给力起飞 / 白灵等编著. —北京:电子工业出版社,2011.9

ISBN 978-7-121-14161-4

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

中国版本图书馆 CIP 数据核字(2011)第 146582 号

策划编辑:张月萍

责任编辑:李云静

印 刷: 北京市铁成印刷厂
装 订:

出版发行:电子工业出版社

北京市海淀区万寿路173信箱

邮编:100036

开 本:787×980 1/16

印张:36

字数:922千字

印 次:2011年9月第1次印刷

印 数:4000册 定价:69.00元



凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至zltz@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线:(010)88258888。

前 言

这是一个编程技术快速发展的时代，这是一个新的编程语言不断出现的时代，而 Java 始终站在这股奔涌大潮的最前列。Java 具有“一次编码、到处运行”的强大特性、针对各种开发领域的平台支持（Java ME、Java SE、Java EE）、免费的开发工具（Eclipse 和 NetBean 等）。此外，作为开源的语言，Java 为广大软件开发者提供了一个开放性的、优秀的开发平台，强大的开源社区和 IBM 等大公司的支持为 Java 的发展提供了源源不断的动力。

Java EE 是 Java 的主要应用平台。虽然面对着激烈的竞争，但 Java 仍然是现在 Web 应用领域中最有竞争力的开发语言。.NET 及 Ruby 等后起之秀都继承了 Java 的一些特点及设计理念，学好了 Java，对于学习其他开发语言和技术也有莫大帮助。

本书讲解了 Java 面向对象编程知识和 Java EE 的主流开发技术，包括 JSP、三大框架（Struts、Spring 和 Hibernate）和 Ajax 等，还对 MySQL 数据库、Tomcat 服务器及 Eclipse 在开发中的使用技巧做了介绍。

本书与普通的 Java 书籍的不同在于：

- 以任务为驱动，让读者充分参与到知识的学习过程中来，不仅看懂，而且会用。
- 不仅讲解各种开发技术“怎么用”，而且分析“为什么要用”，更为读者讲解各种技术的学习方法和经验，让读者学完之后能做到举一反三、“知其然，更知其所以然”。
- 以一个实际系统的开发贯穿始终，而不是零散地讲解一些知识点，让读者对于书本知识在实际开发流程中“什么时候用、怎么用”等问题有一个清晰的认识。
- 使用幽默生动的语言，配以图片讲解生涩枯燥的知识，让读者容易理解、在学习的过程中感到轻松愉快。
- 以“实用性”为最高原则，所讲的知识 and 例子都经过精练，融入实际开发中的技巧，让读者学会之后马上能在实际的开发中用上。

本书的结构如下：

- 第 1 章 介绍 Java 语言及面向对象编程的基础知识，通过在任务中完成一个简单的桌面程序——日记本，讲解 Java 语言特性的应用及 Eclipse 这一开发工具的使用技巧。
- 第 2 章 介绍 Java 数据库编程的相关知识，通过为第 1 章的日记本程序加入数据库存储功能，讲解使用 JDBC 访问 MySQL 数据库的方法和 SQL 语言的基本使用。

- 第 3 章 介绍如何使用 JSP 进行简单的 Web 开发。通过社区系统的注册、登录等简单模块的构建,讲解 JSP 的基础知识及使用方法,在前两章基础知识铺垫的前提下,实现从桌面应用到 Web 应用的过渡。
- 第 4 章 介绍 Hibernate 框架的使用。通过将 Hibernate 整合到社区系统中,讲解 Hibernate 的基本原理和配置方法。
- 第 5 章 介绍 Struts 框架的使用。通过用 Struts 将社区系统改造为一个标准 MVC 结构,讲解 Struts 2.0 的基本原理及使用。
- 第 6 章 介绍 Spring 框架的使用。在任务中使用 Spring 的依赖注入和 DAO 整合等工具优化社区系统的后台程序架构。
- 第 7 章 介绍 Struts、Spring 和 Hibernate 的整合开发,总结并复习之前章节的内容,对社区系统加以完善。
- 第 8 章 介绍 Ajax 技术的应用,使用 Ajax 来改善社区系统的外观和操作等用户体验。

另外,本书 4 个附录涵盖了软件开发环境配置、Jar 包导入和管理、Java Web 开发常见错误的解决之道及常用网络资源等内容。

读者可从网上下载本书的资源包,网址是 <http://www.broadview.com.cn/14161>。

由于作者水平所限,书中出现错误和疏漏之处在所难免,恳请读者见谅并指正。除封面署名作者外,陈其也参与了本书的编写工作。最后,还要对在本书的编写过程中为我们提供无私帮助的朋友们表示感谢。

目 录

第 1 章 Java 面面观	1
1.1 认识 Java	2
1.2 面向对象	3
1.2.1 面向对象语言的诞生	3
1.2.2 建立“一切都是对象”的观念	4
1.2.3 认识对象的模具——类	5
1.3 第一个任务——日记本	8
1.3.1 设计日记本程序的类	9
1.3.2 将类的设计转为代码	10
1.3.3 组装日记本程序	25
1.4 温故而知新	31
1.4.1 本章小结	31
1.4.2 添砖加瓦	32
1.4.3 下一步做什么	38
第 2 章 JDBC	39
2.1 任务分析	40
2.1.1 使用数据库的优势	40
2.1.2 如何使用数据库	41
2.2 JDBC 简单用	43
2.2.1 JDBC 是什么	43
2.2.2 用 JDBC 访问 MySQL 数据库	45
2.2.3 简单的 SQL 数据操作	51
2.3 用 JDBC 处理日记	56
2.3.1 日记的存储	56
2.3.2 日记的读取	60
2.3.3 日记的更新	64
2.3.4 日记的删除	67
2.4 温故而知新	72
2.4.1 本章小结	72

2.4.2 添砖加瓦	73
2.4.3 下一步做什么	78
第3章 JSP 网页设计	79
3.1 从 HTML 到 JSP	81
3.2 社区系统任务开始	84
3.3 深入浅出 JSP	87
3.4 用户注册和登录	104
3.4.1 创建数据库	104
3.4.2 连接数据库	105
3.4.3 添加注册和登录模块	109
3.5 信息发布	138
3.5.1 模块功能介绍	138
3.5.2 想好就动手	138
3.6 温故而知新	143
3.6.1 本章小结	143
3.6.2 添砖加瓦	143
3.6.3 下一步做什么	144
第4章 Hibernate	145
4.1 储藏室管理员——Hibernate 基础	146
4.1.1 Hibernate 的基本工作原理	147
4.1.2 连接对象与数据库表的纽带——映射文件	149
4.1.3 储藏室的基本信息——配置文件	153
4.1.4 这位管理员称职吗	156
4.2 升级信息发布系统	159
4.2.1 信息发布系统的用户界面	161
4.2.2 把信息存储到数据库中	164
4.2.3 信息的表示	166
4.2.4 从数据库中存取信息	176
4.2.5 信息发布系统功能实现	188
4.2.6 将信息发布系统展示给用户	193
4.3 温故而知新	210
4.3.1 本章小结	210
4.3.2 添砖加瓦	211

4.3.3	下一步做什么	211
第 5 章	Struts 2.0	212
5.1	走进 Struts	213
5.1.1	Struts 2.0 架构初探	214
5.1.2	从“Hello, World”开始	215
5.2	由“登录”开始	227
5.2.1	使用 action 的属性来接收用户输入的数据	228
5.2.2	使用领域对象接收用户输入	230
5.2.3	访问 request、session、application 对象	234
5.3	升级信息发布系统和评论系统	239
5.3.1	web.xml 配置文件	241
5.3.2	struts.xml 配置文件	242
5.3.3	Struts 2 中的 OGNL	256
5.3.4	为简易社区系统引入 Struts 标签	258
5.4	温故而知新	285
5.4.1	本章小结	285
5.4.2	添砖加瓦	286
5.4.3	下一步做什么	286
第 6 章	Spring	287
6.1	Spring 介绍	288
6.1.1	Spring 的 IoC 和 DAO	288
6.1.2	第一个 Spring 示例	290
6.2	Spring 核心	302
6.2.1	Spring 基本原理	302
6.2.2	基于 XML 的 Spring 配置	303
6.2.3	实例化 bean	305
6.2.4	依赖注入	308
6.3	升级信息发布系统	317
6.3.1	界面设计	318
6.3.2	数据库和持久层设计	321
6.3.3	DAO	331
6.3.4	业务逻辑层	335
6.3.5	Spring 的配置	340

6.3.6 最后的整合	348
6.4 温故而知新	350
6.4.1 本章小结	350
6.4.2 添砖加瓦	351
6.4.3 下一步做什么	351
第 7 章 Struts+Spring+Hibernate 开发简易社区	352
7.1 社区整体架构	353
7.2 信息发布系统	389
7.3 评论管理系统	431
7.4 用户管理系统	449
7.5 温故而知新	469
7.5.1 本章小结	469
7.5.2 添砖加瓦	469
7.5.3 下一步做什么	469
第 8 章 Ajax	470
8.1 Ajax——“掀起你的盖头来”	471
8.1.1 结合实际了解 Ajax	471
8.1.2 试试效果	472
8.2 注册不再烦人	483
8.3 翻页也提速	492
8.4 社区提速总体验	504
8.5 温故而知新	517
8.5.1 本章小结	517
8.5.2 添砖加瓦	517
8.5.3 海阔凭鱼跃，天高任鸟飞	517
附录 A 软件开发环境配置	519
附录 B Jar 包导入和管理	554
附录 C Java Web 开发常见错误的解决之道	558
附录 D 常用网络资源	566

第 1 章

Java 面面观

已经退休的比尔·盖茨曾经说过，它是长期以来最好的编程语言。

据统计，它的相关产业占据了印度 GDP 的 2%。

TIOBE 每月发布的世界编程语言排行榜上，它总是坐在第一的位置，没有半点悬念。

这就是 Java，从 1995 年发布至今，它刮起的热潮从未消退。这个最初为电视盒等小型设备设计的编程语言，如今已经成为网络应用领域当之无愧的王者。

Java 的名字据说起源于 Java（爪哇）咖啡。学习 Java 语言就像是品一杯咖啡一样，除了享受快乐的香浓，味蕾也要做好挑战苦涩的准备。希望本书能为您沉淀咖啡的苦，留下丝丝芬芳和回味无穷。

端起这杯咖啡，本书即将进入 Java 的神奇世界。在学习本书的过程中，通过一系列简单有趣且富有挑战性的小任务，循序渐进，读者将会逐步完成一个完整的社区系统。在系统的建造及完善中，读者也逐步学会 Java Web 开发的相关知识。

在本章中，读者将会对 Java 有一个初步的认识。本章的任务，是一个简单实用的日记本程序，读者只需根据提示一步一步地完成它。在此过程中，读者将对 Java 基础知识的实际应用以及开发工具的使用有一个初步的体会。好了，祝您的第一杯咖啡喝得愉快。

1.1 认识 Java

学习一门编程语言之前，必须对它的背景、用途等方面有清楚的认识。编程语言和汉语、英语等沟通语言一样，要和具体的应用环境结合起来才能发挥作用。试想一下，一个人费尽千辛万苦学会了传说中的火星语，却发现这个世界上没有火星星人（如图 1-1 所示）……

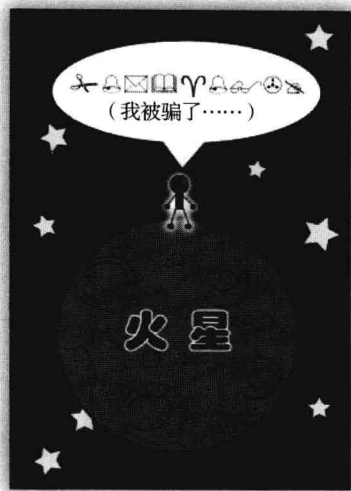


图 1-1 回火星去吧

1. Java 到底能干什么

许多 Java 初学者都会有这样的疑问。从理论上讲，其他编程语言能做的事情，Java 基本上都能够做，但是“能做”并不意味着“适合做”。如果要开发一个微软的 Word 之类的桌面应用程序，或者是魔兽世界这样的大型游戏，Java 绝对不是理想选择。所以，“Java 到底能干什么”应该换成另一个问题。

2. Java 适合干什么

“Java 不只是一门语言，而是一个平台。”接触 Java 的人一定都听过类似的说法。这句话是什么意思呢？所谓平台，可以理解为一个工具箱，里面的工具可以代替人完成很多复杂的任务，从而降低工作难度，减轻工作负担。Java 之所以成功，并非因为它本身是一门语法优美、设计先进的语言，而是由于它提供了丰富而强大的工具箱：SE、EE 及 ME（如图 1-2 所示）。这三大平台面向不同的领域，其中得到最长足发展和广泛使用的，无疑是 Java EE（Java Enterprise Edition，Java 企业版）。Java 提供的强大的网络操作、安全等特性和独一无二的“一次编写，到处运行”的跨平台能力，使其在面向网络的企业应用、开发、集成等方面如鱼得水。

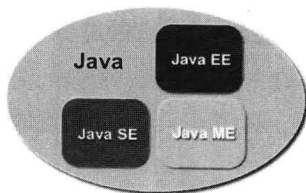


图 1-2 Java 平台

面向网络的应用，特别是 Web 开发，是适合 Java 大展拳脚的世界。曾使用过其他语言进行网络编程的人，在使用 Java 之后都会感动得一塌糊涂：因为 Java 把建立连接、发送消息等烦琐的网络任务简化到了极点。这是一个网络时代，而 Java 就是为网络而生的。

1.2 面向对象

1.2.1 面向对象语言的诞生

编程本质上是和计算机打交道，用程序去控制计算机的行为，从而解决现实世界中的各种问题。但计算机是个极其复杂的系统，并不是每个编写程序的人都需要打开电脑机箱，搞清楚它的内部结构和工作原理（如图 1-3 所示）。

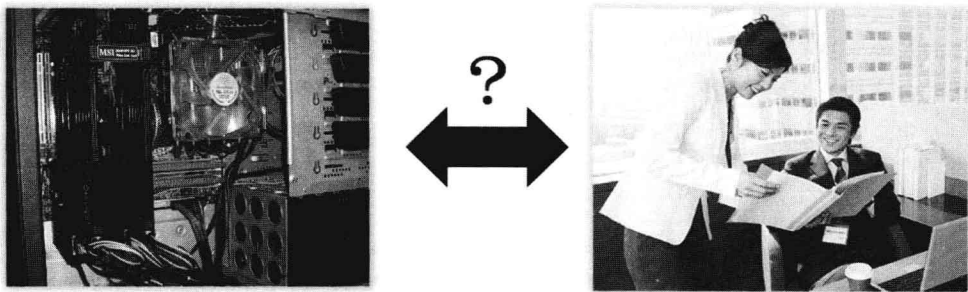


图 1-3 计算机与程序员

将计算机的机器细节变成人脑容易理解和记忆的形式，将计算机的世界和人类日常生活的世界联系起来，这就是编程语言所要完成的任务。编程语言的发展就是从计算机世界向现实世界，从计算机思维向人类思维不断靠拢的过程。

计算机的世界中只有 1 和 0 两个数字，最早的“编程语言”是在长长的纸带上用打孔和不打孔来表示 1 和 0，以此来操纵计算机的。这样的方法费时费力费资源，一个简单的程序需要的纸带就多到令人发指的地步。此外，它还很容易出错，一个孔打错就意味着一整条纸带的报废，让人欲哭无泪。

纸带时代之后，汇编语言出现了，写程序不再等于打孔，写程序的人不必再用 0 和 1 思考，一条

条的汇编指令代替了让人抓狂的纸带。这是一个巨大的进步。

汇编语言之后，出现了 BASIC 和 C 等高级语言，复杂的汇编指令消失了，程序代码在可读性、可维护性等方面都有了很大提升。可是，程序语言仍然是贴近计算机逻辑的东西，离人类思考问题的模式尚有很大的距离。

从打孔纸带到汇编指令，再到高级语言中的函数，接下来，面向对象语言出现了，Java 就是其中的典型代表，它的口号是：一切都是对象（如图 1-4 所示）。

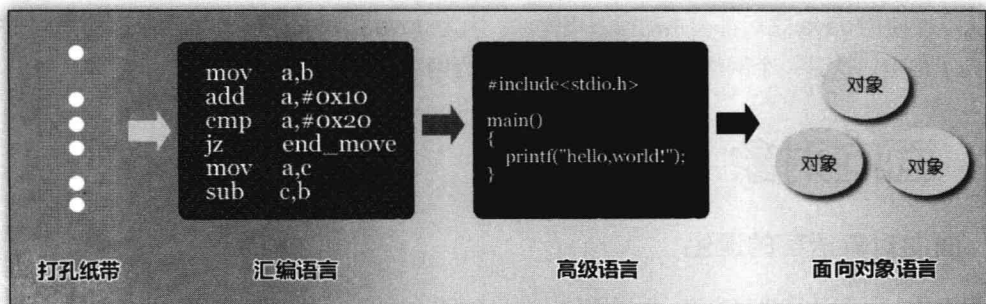


图 1-4 语言的发展

1.2.2 建立“一切都是对象”的观念

“面向对象”的英文原名叫“Object-Oriented”，简称 OO，根据翻译的不同，它也有其他不同的称谓，如台湾地区惯用“物件导向”这种叫法。在 Java 的世界中，所有的东西都有一个统一的称谓——对象。对象是什么呢？简单说来，对象=属性+方法（如图 1-5 所示）。

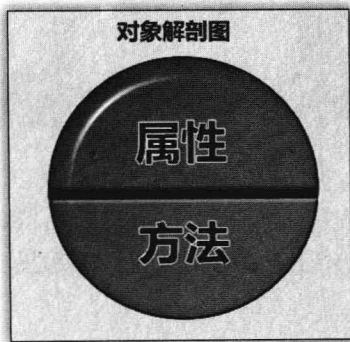


图 1-5 对象

属性也可以称为特性，它从静态上描述了该对象的各种特性；方法也可以称做行为，它从动态上描述了该对象能够做的事情。例如，一个人的属性可以包括姓名、年龄、性别等，一个人的行为可以

包括思考、说话、吃饭、睡觉等。

理论上，人们脑海中的任何东西都能成为程序中的对象。小到一粒沙，大到一个宇宙都可以是对象；不仅包括实实在在的东西，就连“思想”和“灵魂”这些抽象的概念也可以变成对象——总之，只要能用属性+方法的形式来描述的东西，都可以成为对象（如图 1-6 所示）。

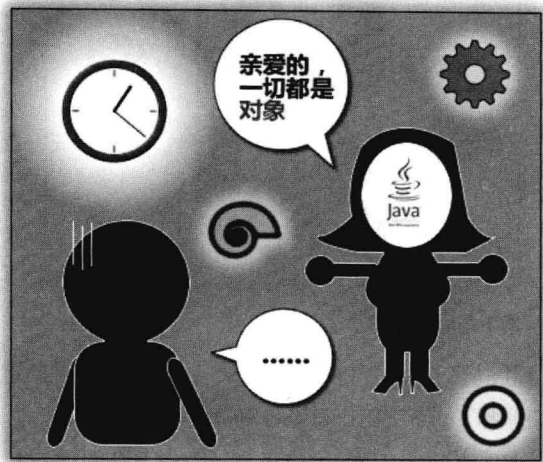


图 1-6 面向对象的 Java

了解了对象的概念之后。在面向对象的世界里，所谓的“程序”就不再是一条条的命令，而是成堆的对象组合在一起，通过彼此间传递消息互相协作，从而完成特定的任务。例如“上课”这个简单的事情，便可以转化为“学生”对象和“老师”对象之间的关系。这和现实世界是一致的。与 BASIC、C 这些高级语言相比，Java 这样的面向对象语言更加符合人类认识问题、思考问题的模式。

1.2.3 认识对象的模具——类

以“上课”为例，这个例子中同班的学生们除了姓名、年龄、性别等属性不同之外，其他属性是完全相同的，所以他们都属于同一“类”。

“类”是面向对象程序设计中一个极其重要的概念（如图 1-7 所示）。读者已经知道，对象包含了属性及行为。“类”便是有着共同属性及共同行为对象的总称。在面向对象的 Java 程序设计中，所有的个体都是对象，每个对象都属于特定的类，类定义了属于该类对象的属性及行为。例如属于“学生”这个类的对象，都具有姓名、年龄、性别等属性，只是这些属性的值不一样。除了属性之外，他们还具有“记笔记”等“学生”类所定义的行为。

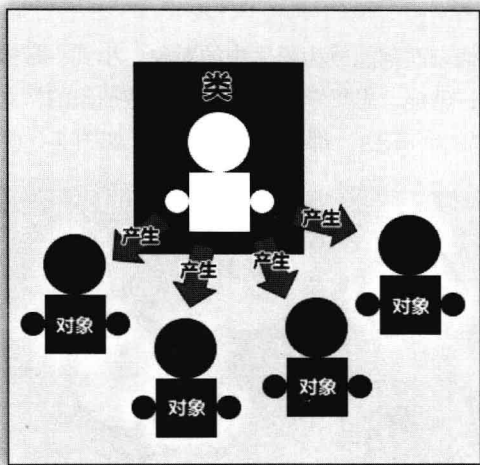


图 1-7 类和对象

类就像是对象的模具，程序员只要设置好了类，那么属于该类的对象想产生多少就产生多少。这个产生的过程称为类的“实例化”，所产生出来的对象，称做类的实例。

在 Java 里面，用关键字 **Class** 来表示类。前面所说的“学生”类名字是 **Student**，它有 3 个属性：

- (1) **name**: 姓名，字符串类型。
- (2) **age**: 年龄，整数类型。
- (3) **sex**: 性别，字符串类型。

因此，这个“学生”类——**Student**，可以用 Java 代码描述如下（参见清单 1-1）。

清单 1-1

```
class Student {
    String name;    // 姓名
    int    age;     // 年龄
    String sex;     // 性别

    /**
     * 学生上课记笔记
     */
    void takeNote(){
        System.out.println("我是" + name);
        System.out.println("今年" + age + "岁");
        System.out.println("性别" + sex);
        System.out.println("我开始记笔记了");
    }
}
```

```

    }
}

```

还没有为这个类创建构造函数（这种情况下，Java 编译器会自动为其创建一个无参数的构造函数），它只有一个叫做 `takeNote` 的方法，表示上课记笔记这个行为。方法中使用了 `System.out` 对象，调用它的 `println` 方法，向控制台输出信息。

Java 命名规范

关于类、属性和方法的命名，满足以下两点的名字都是合法的：

- (1) 以字母、下划线及美元符号“\$”开头，随后是字母和数字的任意组合。
- (2) 非 Java 的保留字（如 `class`、`void` 等）。

按照 Java 的命名规范，通常情况下，类名的第一个单词以大写字母开头，属性名、方法名的第一个单词以小写字母开头，其后单词——如果有的话，均以大写字母开头。类名和属性名应该是一个名词，方法名应该是一个动词或动宾短语。在清楚表达意思的前提下尽量简短。

为了测试这个类，读者需要为它加上一个 `main` 方法。`main` 方法是每个 Java 应用程序都必须有的，每一个 Java 应用程序都是由 `main` 方法开始的，而 `main` 方法执行完毕就意味着该应用程序的结束。

添加了 `main` 方法之后，`Student` 类变成了下面这个模样（参见清单 1-2）。

清单 1-2

```

class Student {
    String name; // 姓名
    int    age;  // 年龄
    String sex;  // 性别

    /**
     * 学生上课记笔记
     */
    void takeNote(){
        System.out.println("我是" + name);
        System.out.println("今年" + age + "岁" );
        System.out.println("性别" + sex);
        System.out.println("我开始记笔记了");
    }

    public static void main(String[] args) {
        // 建立一个名为 zhouxingxing 的 Student 对象
    }
}

```

牢记 `main` 方法的书写，注意 Java 是区分大小写的，如 `main` 写成 `Main` 之类的疏忽都会导致错误。

```
Student zhouxingxing = new Student();

// 为该对象的属性赋值
zhouxingxing.name = "JavaWeb 爱好者";
zhouxingxing.age = 20;
zhouxingxing.sex = "男";

// 调用该对象的 takeNote 方法
zhouxingxing.takeNote();
}
```

在 `main` 方法中，读者可以看到，这里通过关键字 `new` 创建了一个叫做 `zhouxingxing` 的 `Student` 类型的对象，然后对它的各项属性进行赋值，姓名设为“JavaWeb 爱好者”，年龄设为 20 岁，性别设为男。最后，调用他的 `takeNote` 方法，让这个学生记笔记。

运行这个简单的程序，就可以在控制台上看到如下输出：

```
我是 JavaWeb 爱好者
今年 20 岁
性别男
我开始记笔记了
```

以上通过代码展示了一个简单类的创建以及使用——与其说简单，不如说它非常的简陋。如果读者具备 **Java** 编程的基础知识，可以很容易地指出其中的问题：比如，类的属性应该是对外部隐藏的，对属性的访问和修改应该通过标准的 `get` 和 `set` 方法；类及其属性、方法等应有明确的访问修饰符；等等。本书的重点在于讲解应用开发，但是牢固的基础知识和严格的编程规范是极为重要的，一些读者写了很多的代码，编写了很多的程序，但对一些基础的知识却不甚了解，这是本末倒置的做法，对自己的水平是不利的。“万丈高楼平地起”，扎实的基本功是写出好程序的基础。在今后的知识讲解中，笔者将会把重要的基础知识融入其中，请读者牢记它们。即使曾经学习过，也请再温习一遍，说不定会发现以前不曾注意到的东西。

有些没有使用过 **Java** 开发工具的读者可能会无法去比对，不过不用着急，这里主要是向读者示范。在下一节，笔者会详细介绍开发工具的使用。

1.3 第一个任务——日记本

单纯地学习一门语言不是本书的目的，将知识转化为实际的运用才是笔者的意图。现在开始本章的实践任务——日记本程序。

也许读者会问，最终要完成的不是一个 **Web** 社区系统吗，做这个日记本有什么用呢？因为 **Web**