



喻桃阳 编著

数据库开发这点事

Java 快速入门

— 人生需要引导系列丛书 —

全球第一本启用思想引导技术快速研发的书

中国第一本从入门到快速融通、并直接应用的书

新老程序员读阅此书，打破靠吃青春饭搞研发的说法

聚引导学思想及实际案例与作者十年经验，内容通俗易懂

浩为教材

内容摘要

代码只是一个过程，编程是一种思想，项目开发更是一个系统工程。对程序员来说，我想绝大多数绝不会只期望自己写一辈子代码，而是希望编码几年后就不再编码或少编码，成长为更高一级的 IT 人士。所以在编程时就得培养自己的整体思想，在学习时更应该去学习整体思想。

从职业发展来说，大多程序员先是只会写一点代码，随着经历的积累，慢慢掌握系统工程的思想，然后能写一些简单的文章，再做一些需求分析，再系统设计……也就是说，程序员最后都会掌握思想，既然如此，何不在一开始就在这种思想的指导下学习呢？

比如对 Java 基础的学习，在掌握三大基础的情况下，再去查看 JDK 的 API 文档，搞清每个常用包的作用（即在你的脑海中建立一个模型，可用 FreeMind 自己做一个常用包的归纳图）就可以，用不着在每个类上花时间。学以致用，当你要用时，再去详细看一下 API 或 Google 一下，很快就掌握并能用了。

摘自“10.8 编程更需要引导：学以致用”

数据库原理，讲述了一堆又一堆的理论，一看就让人头痛。但对绝大多数人来说，简单概括如图6-8所示，数据库就分为数据库对象和数据库操作，而其他知识，除非要去进行这方面的研究，没有多大必要去浪费时间。

摘自“6.4 数据库模型”

有个将在 2011 年考研究生的大三学生问我“考研复习，每天的各个时间片和对应要完成的任务可不可以做成思维导图？”我简单说了一下，他就明白了，试着画了一张，然后截图给我看。以下是整理后的对话：（他签名为“天行健，君子以自强不息！”，简称为自强）

自强：感觉就用到几个功能，先把大体轮廓做好了，我感觉应该往后细分，越详细越好。

浩为：感觉到思维导图的便利性了吧，往后细分后抓出重点。

自强：一个考研成功的师哥告诉我习惯的重要性，拿出点时间来制定计划，肯定比盲目往下学有好处。

浩为：不仅是考研，以后的工作、生活更是如此。

自强：暑假的时候，就应该学会这个东西。（注：他曾参与浩为公司在暑假发起的“软件工程术语中英对照案例解析教程”项目）

浩为：现在学会还来得及。

摘自“2.3 FreeMind 应用范例”

附：

营销策划：刘翼炜先生 [互联网全球展览中心CEO \(hlwqz.com\)](http://hlwqz.com)

提供素材名单：陈聚雄、许童童、高为明

人生需要引导IV

喻桃阳 著



数据库开发这点事

Java 快速入门

浩为资源堂，学以致用，以 IT 技术、职场体验为主的资源圈

目录

内容简介.....	1
导读.....	1
IT简单化.....	3
编程核心是代码，还是思想.....	4
学习小组能给大学生带来什么.....	7
学以致用：高为明对话浩为及感想.....	9
苦学勤练——技术高手的进阶之道.....	11
第1篇 入门篇.....	13
第1章 搭建开发环境.....	14
1.1 Java运行环境JRE.....	14
1.2 Web服务器.....	14
1.3 Java开发整合包使用说明.....	16
1.4 Eclipse使用简介.....	19
1.5 新建Eclipse项目.....	21
1.6 新建Tomcat项目.....	25
第2章 激发你的思维潜能.....	28
2.1 我的思维导图使用历程.....	28
2.2 思维导图.....	29
2.3 FreeMind应用范例.....	31
2.4 学会Google快速查找知识.....	32
第3章 HELLO WORLD范例.....	32
3.1 输出单一字符串.....	33
3.2 执行代码.....	34
3.3 输出多个字符串.....	36
3.4 调试代码.....	39
第4章 数据类型.....	40
4.1 计算机数据类型.....	40
4.2 常见数据类型.....	41
4.3 定义常量与变量.....	45
4.4 数值型数据的混合运算.....	49
4.5 算术运算.....	52
4.6 比较和条件运算.....	54
4.7 逻辑和位运算.....	54
4.8 递增和递减与赋值运算.....	58
4.9 计算机基础：原码、反码、补码.....	61
第5章 控制结构.....	62
5.1 条件控制.....	63
5.2 循环控制.....	67
第6章 基于数据库的HELLO WORLD.....	71
6.1 数据操作种类.....	72

6.2 申请资源.....	75
6.3 常用数据操作.....	75
6.4 数据库模型.....	76
6.5 释放资源.....	77
6.6 小语句解决大难题, IT需要简单化.....	78
第7章 人生没有救世主, 一切靠自己.....	79
7.1 商业计划书的启发.....	80
7.2 商业计划书待解决问题.....	80
7.3 自我经历: 知道自己要做什么.....	82
7.4 谋事在人、成事在天, 学会资源整合.....	85
7.5 人生没有救世主, 一切靠自己.....	87
第8章 认识开源数据HSQLDB.....	90
8.1 hsql运行模式.....	90
8.2 hsql基本概念.....	95
8.3 hsql自启动.....	97
第9章 SQLEXPLOER使用简介及SQL简介.....	98
9.1 安装SQLExplorer.....	98
9.2 配置SQLExplorer.....	99
9.3 使用SQLExplorer.....	102
9.4 SQL基础.....	106
9.5 数据操作-针对记录.....	107
9.6 数据定义-针对对象.....	108
9.7 数据控制-授权.....	108
9.8 MySQL快速入门.....	108
第10章 了解JSP.....	112
10.1 初识JSP.....	113
10.2 JSP版Hello World.....	113
10.3 数据库版Hello World.....	115
10.4 HTML语法.....	116
10.5 快速掌握JSP.....	119
10.6 JavaScript语法.....	126
10.7 CSS语法.....	129
10.8 编程更需要引导: 学以致用.....	132
第11章 JSP+JAVABEAN.....	135
11.1 JavaBean版Hello World.....	135
11.2 运营商管理页面.....	138
第12章 数据结构.....	139
12.1 基本概念.....	139
12.2 基础数据结构.....	141
第13章 JSP+JAVABEAN+SERVLET.....	144
13.1 Servlet版Hello World.....	144
13.2 应用系统中的Servlet.....	145
第14章 电话外呼系统.....	147
14.1 硬件配置方案.....	148

14.2 软件部分.....	149
14.3 安装准备.....	149
14.4 系统安装.....	150
14.5 用户种类及功能模块.....	151
14.6 用户操作.....	152
第 15 章 HwCALL 系统设计	157
15.1 可行性分析.....	157
15.2 需求分析.....	158
15.3 总体设计-系统设计.....	158
15.4 详细设计-功能模块.....	158
第 16 章 HwCALL 编码	158
16.1 系统登录详解.....	159
第 17 章 IT 更需要引导.....	164
17.1 IT 培训重在引导.....	165
17.2 引导学思想开创 IT 培训新模式.....	167
17.3 引导学概述：人生需要引导.....	169
17.4 人生需要引导系列总序.....	170
17.5 作者：喻桃阳简介.....	172
17.6 学以致用开创大学社区新模式.....	173
17.7 聚餐有感：大学生该注意三点.....	174
17.8 寄语大学生：社会讲究利益.....	175
17.9 《从人生需要引导论 Java 快速开发》章节说明.....	176

内容简介

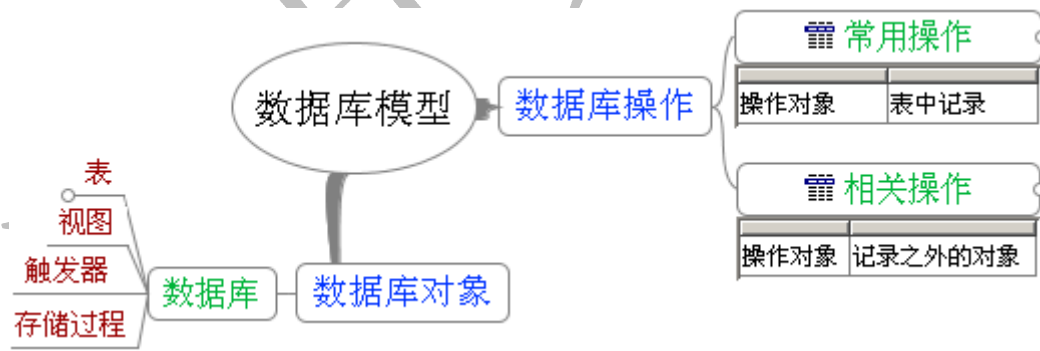
针对 Java 初学者及入门者，系统阐述 Web 开发基本知识，结合数据库的快速应用，使读者快速跨入 Java 领域，对编程语言不再畏惧。该书作为《从人生需要引导论 Java 快速开发》入门版，也是《人生需要引导》系列中技术入门的第一本，已计划撰写《数据库开发者点事-PHP 快速入门》和《数据库开发者点事-C/C++快速入门》，且提供电子版下载。

《从人生需要引导论 Java 快速开发》原名“数据库开发这点事”，在撰写及推广过程中，常和一些大学生交流，感觉他们不仅仅需要技术，更需要人生引导。于是该书定位于准备从事 IT 的初学者及 Java 入门者，既能让初学者快速入门、掌握实用技术，又能让入门者迅速树立整体思想。更让所有读者从根本上明白自己需要什么，懂得资源整合、学以致用，快速开启财富之门。注：资源整合等概念在《喻世明言新篇-人生需要引导》中有全面介绍。

编程的核心是思想，语言只是思想的一种实现方式而已；当你有了思想，即使不懂某种语言，但你很快就能用其实现功能。不论是何种语言：Java、C++、PHP，还是数据库代码，基础知识均为：数据类型、数据结构、控制结构。大部分代码就是这三种内容的组合，再根据软件工程的思想来实现功能。

数据库开发，是企业应用的基础。浩为采用参照现有项目，自己动手做项目；通过通俗易懂的描述，图文并茂，带你畅游数据库。代码只是一个过程、编程是一种思想、项目开发更是一个系统工程，浩为先讲述过程、塑造你的思想，让你树立系統工程的理念。

数据库究竟是什么，详细内容可浏览有关数据库原理的书籍，很厚的一本书，一大堆理论，很快就会让人头大。简单归纳一下呢，数据库就是如下图所示的一个模型，使用 SQL 语句操作数据库对象而已。



你现在想做什么？必须做什么！你以前做过什么？现在能做什么！以后要做什么？现在怎么做！认真考虑考虑这些问题，只会对你百利而无一害。个人成长可以分为多个阶段，每个阶段都存在前面提到的 6 点，引导学将辅助你思考这些问题。教是为了不教，学以致用，引导学更激发你的潜能。有人说作者饱经沧桑，希望其经历及努力能帮助更多的人。

导读

为啥要写导读？导读导读，就是引导如何阅读。该书到底讲了哪些内容，这是大多数读

者最关心的问题；而我最关心的是，这本书到底能给你带来什么？

重点内容推荐：

一、编程的核心是思想，语言只是思想的一种表达方式而已

编程的核心是思想，语言只是思想的一种实现方式而已；当你有了思想，即使不懂某种语言，但你很快就能用其实现功能。不论是何种语言：Java、C++、PHP，还是数据库代码，基础知识均为：数据类型、数据结构、控制结构。大部分代码就是这三种内容的组合，再根据软件工程的思想来实现功能。

二、人生定位：人生需要引导，知道自己要做什么

社会需要什么、企业需要什么、你需要什么，还没有正确定位的你，该怎么样确定自己的方向。定位之中就包含职业定位，这需要自我改变，但自我改变是个艰难的过程，需要外力的协助。人生需要引导将需要引导的人组成一个群体（浩为资源堂 zyt.howwe.net），辅以日程事务管理，携手共进，共同发展。你现在想做什么？必须做什么！你以前做过什么？现在能做什么！以后要做什么？现在怎么做！认真考虑考虑这些问题，只会对你百利而无一害。



有人说，此 Logo 下半部分像一个碗，那人生需要引导就是顶部的光源，让你的碗快速充实，同时刻上 Howwe 的烙印。

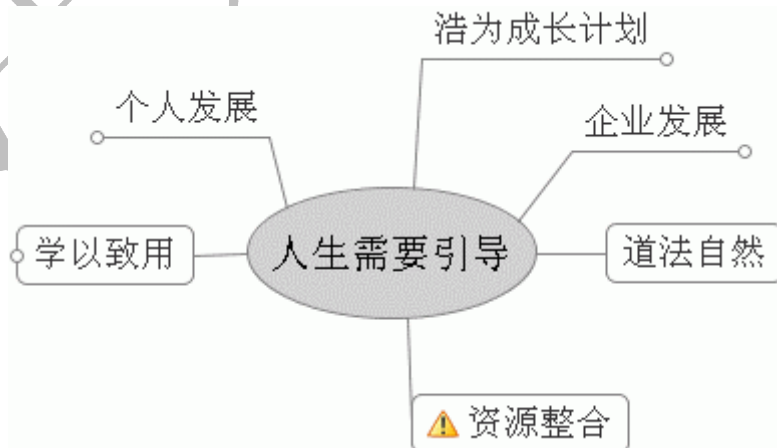
人生没有救世主，一切靠自己。很多人总幻想别人会帮你，可别人为啥要帮你，你什么地方值得别人来帮？现实一点吧，一切靠自己，先从小事做起，更不要幻想能一步登天。

从系统论的角度来看：人生不会有所谓的最终的成功，有的只是你对人生的豁达和你看事物的那份坦然。人活的是一份良知和一份责任，活的是一种平凡和一种心态！

三、资源整合，学以致用

教育的目的是什么，很多人在思考，更有不少人在反思。招聘方和应聘方粗看起来是一对矛盾体，可哪个招聘者不希望又快又准地给公司找到合适的人选，又有哪个应聘者不希望能尽快通过面试获得工作机会、而完成自己职业生涯的又一次跃进呢？

下图为人生需要引导（简称引导学）相关概念的思维导图：



1、个人的发展需要教育，更需要引导；企业的发展需要人力资源，更需要自我发展。教育怎样才能把两者结合起来，学以致用是根本。

2、个人发展存在很多不可控因素，道法自然让人坦然面对。

3、资源整合，对个人和企业同等重要，学习型组织更体现了对个人的资源整合。

4、从系统论的角度来看：人生不会有所谓的最终的成功，有的只是你对人生的豁达和你看事物的那份坦然。人活的是一份良知和一份责任，活的是一种平凡和一种心态！

中国有句古话“盖棺论定”，这也是对中国几千年来成功人士的一个概括和告诫，如果你真想追求那种最终的成功，就脱离不了“盖棺论定”的宿命。

学习型组织的不断发展，为企业的壮大提供了动力，也为个人的发展提供了机会。浩为倡导学以致用，将激发你的潜力，为你提供发展动力。

微软推出了搜索品牌 Bing，有人说这是“画虎不成反类犬”。尽管 Bing 的中文宣传叫“必应”，但 Bing 这几个字母，总让人觉得别扭，拼音 bing 对应的汉字在国人心中大多为“病”，因为中国人被病给吓怕了，更被治病的高昂费用给吓怕了。

Bing、病，从长远来看，每个人都有病或有病。不过，至少我暂时没病！中国文化的博大精深，将会让微软这个外来者吃大亏，因为“病”犯了国人的大忌。可见定位的重要性，人生需要引导的重要性。

IT 简单化

IT 专业技术是进入 IT 行业的“敲门砖”，要想成为 IT 人士究竟需要掌握哪些技能呢？这方面的书籍数不胜数，不少书中的内容更让读者迷惑。读者感叹：IT 怎么这么难。

先看看有人提出的一个《软件程序员专业技能自检表》，有少许修改。

软件程序员专业技能自检表				
	基础项	熟练掌握？	有项目经验？	列入学习计划？
操作系统	Windows	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Linux	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	UNIX	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
数据库	SQL Server	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	ORACLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	DB2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
基础知识	HTML 语言	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	XML 语言	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	数据结构常用算法	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	UML 知识	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
开发语言和工具	VC++	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	C++	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	VS. NET	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Java	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	其他请注明	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
软件过程	软件工程理论	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	质量体系	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	过程控制	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	软件设计	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	单元测试工具	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	软件加密	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	源代码控制	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	说明书编写	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	程序部署	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
项目组 建设	编码规范	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	源代码控制规范	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	数据库设计规范	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	公共组件或基类	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

那人提到：测试结果怎么样？如果所列的技能大部分你都已经熟练掌握，那么恭喜你，你是一名合格的程序员！如果这些技能大部分你都已经拥有了项目经验，那么更要恭喜你，你是一名优秀的程序员！如果都没有，那么赶快把他们列入你的学习计划，跟随本书的学习，你很快就会成为一名真正的程序员！

我一测试，长叹一声，我这 10 多年白挨踢了，按他的说法，估计我还算不上程序员？

IT 真有这么难吗，以前有不少学习 JSP 的人问我，按照书上的介绍，他们不论怎么配，也没法搭建开发环境。我一问过程，书上说要装 JDK、Mysql、Tomcat、Resin，甚至还要装 IIS、Apache。有必要这么复杂吗？对于一个初学者，不晕才怪。写书的人是这样，厂家更是这样。浩为，依托多年的技术积累及高校的研发实力，致力于 IT 简单化，坚持学以致用，为广大 IT 人士、或立志于从事 IT 的人士提供 IT 简单化教程及技术。

数据库开发，是企业应用的基础。浩为采用参照现有项目，自己动手做项目；通过通俗易懂的描述，图文并茂，带你畅游数据库。代码只是一个过程、编程是一种思想、项目开发是一个系统工程，浩为先讲述过程、塑造你的思想，让你树立系统工程的理念，从而实现你、实现浩为的目标。编程的核心是思想，语言只是思想的一种实现方式而已；当你有了思想，即使不懂某种语言，但你很快就能用其实现功能。

社会需要什么、企业需要什么、你需要什么，还没有正确定位的你，该怎样确定自己的方向？教育的最终目的是为了应用，学以致用是实现自我发展的根本。

在现实社会中，招聘方和应聘方粗看起来是一对矛盾体，可这两者何尝不是一种合作关系呢？哪个招聘者不希望既快又准地给公司找到合适的人选，又有哪个应聘者不希望能尽快通过面试获得工作机会、而完成自己职业生涯的又一次跃进呢？

对于学生来说，读书 N 年，耗费了亲人的巨额资金，快速掌握一门技术，走上工作岗位，既是对自己能力的肯定，更是对亲人的感恩回报。通过本书的学习，既能让你快速掌握数据库开发的知识，更将为你打造一种思维能力及习惯。

编程核心是代码，还是思想

许童童 090517 撰写 陈聚雄及浩为喻桃阳修改

（注：在交谈内容中“浩为喻桃阳”简称为“浩为”）

昨天在群里跟浩为发生了点分歧，貌似有点让浩为不愉快，在此我说声对不起！

浩为语：讨论有分歧很正常，如果这点小事都不能容纳，那气量也太小了。由于思维的局限性，当一个人处于思维定式中时，正面很难说服，旁敲侧击一般很快就能见效。

我仔细分析了一下原因，从我的经历来看，我很看重代码，觉得编程思想是可以从代码

中提取。在群里一味强调代码重要，称得上是误人子弟！实在太不应该。还傻呵呵地问“思想怎么才能提炼出来”，只怪自己没能看透《人生需要引导》！其实此书已经明确提出，“思维离不开社会实践，它是在社会实践中产生，并发展起来的”。可见思想能指导社会实践，并在实践中得到提升。

聚雄语：其实这也是一种方式，但角度与浩为所说的不一样。浩为是先有思想，后有代码实现。而你是通过读取已有代码，了解其为什么这样写，从而为自己下一次编写代码提供思想，其实都是从思想出发，只不过浩为的方式会少花时间，也许这才是编程之道。

一个只会写代码的程序员，很难甚至永远当不了系统架构师。

原因在于：代码只是软件产品的小部分，只是实现的一种方式，而思想才是产品的精华！系统架构师需要的是思想，解决问题的思想。也许让系统架构师去写代码，他不一定写得出来，但即使你能写出所有代码，根本不能说明你比他强。

为什么他是系统架构师，而其他人不是？难道仅仅因为其他人写的代码比他多？我以前从没想过这些，因为我的接触面很有限，成天写代码。写的代码一多就开始飘飘然，以为自己有思想了，其实我做的那些东西在别人眼里也许一文不值，甚至连半成品都不是。却把这些当成飘飘然的资本，哎！只怪自己像井底之蛙，看到得天空就那么一小片！真要看到整个天空，就必须跳出这个井，扩大自己的知识面，让自己变强！在这过程中要想少走弯路，就需要高人指点。

程序员道路坎坷，想想我初学 Java 时，程序调不出来，就自己一个人在想，却没想在自己眼里的大麻烦，在别人眼里却只是一个小错误，那么微不足道。当自己绞尽脑汁都不想出来时，为何不向别人求助呢？当然可以在别人给你调程序的过程中多学习，学习人家调程序的方法，以免自己总有求于人。

大一时我犯了一个致命错误，因为以前从没接触过计算机语言，而 C++ 讲了一堆概念，什么数据类型、语句、对象，根本就不知是啥玩意，一听就蒙，感觉老师在讲天书。那时就觉得自己能看懂代码，写不出代码是件好事，至少是件不坏的事，却忽视了很重要的一点：写代码是要思想来支撑的，为什么我写不出代码，就是因为自己没有思想。

你能看懂代码，仅仅好比能把英文著作按字翻译成汉语，但你翻译出来的文章有你的感情吗？仅仅只是简单的体力劳动而已，这是很可怕的事情。可以想象，这样翻译的书会有多枯燥，会让读者感觉到他在读字。这不是做无用功吗？所以翻译得把自己的感情加到里面！对于程序来说，如何把思想融入到程序中，也得依靠思想来指导代码，而不是从代码中产生思想。

以前学习时，觉得要读懂程序中的思想，先把那些代码运行一遍，看看结果，看看跟自己想要的结果是不是一样，不一样就问为什么？再把自己想的那些代码写到程序中，慢慢去改变原来的程序，然后就是不断调程序！等程序调出来了，很兴奋！就这样我觉得我有思想了，以为“思想是从代码提炼出来的。”其实“代码只是一个过程、编程是一种思想、项目开发更是一个系统工程”。

以下是浩为在帮我改变观念时使用的语句：

苹果是大家都熟悉的一种水果，我们最初是看到这种水果，知道它的形状、颜色等特征，它还能吃；然后有人告诉你，这个是苹果；在学英语时，有人教你这个叫 Apple。

在这过程中，看到实物后认识是形象思维，别人告诉你它叫“苹果”或“apple”是逻辑思维。在我们没有这个概念前，如果别人只教你“苹果”这个词，一般很难记住。当你看到苹果或告诉你苹果是什么后，很快就记住了，因为我们脑子里就会有苹果的模样。其实我

们一说到苹果，一般先现出苹果的形象，再经过逻辑转化把苹果和苹果的形象关联起来。

只讲代码，好比是告诉你一堆词组，这些词组是你从来就没有见过的，你能记住吗？而思想结合代码，很快就能了解，并记住了。纯代码就是抽象的东西。

什么是形象思维？形象思维是运用头脑中积累起来的表象进行的思维。表象是我们以前知觉过的，而在头脑中再现的那些对象的映象。即现实中的苹果或描叙的苹果是对象，通过形象思维处理后，在我们的大脑中形成苹果的映象。

什么是逻辑思维？逻辑思维是人们在认识过程中借助于概念、判断、推理等思维形式能动地反映客观现实的理性认识过程，也称抽象思维。即用“苹果”这词和形象思维形成的苹果的映象对应起来。

如前所述，抽象思维与形象思维是两种基本的思维方式，人类从事各种活动，往往需要对两种思维方式协同使用。专家的大脑中有着丰富的形象贮备，在解决问题时，他们总是先根据问题情景构建出清晰的物理图象；尽可能利用图形来反映物理状态、物理过程及有关物理量的关系；善于在头脑中对有关形象进行分析、比较、类比、整合；想象可能发生的情景。所以，专家往往对问题的形象有着较强的直感能力。而一般人的大脑中，形象的贮备相对贫乏，他们在解决问题时，不善于从形象上去把握问题；一接触到问题，就企图立即给出解决方案，其结果往往是欲速则不达。

我们先从最简单的代码入手，输入一个字符串“Hello World”，而字符串就是一种数据类型。再输出多个。为啥能输出多个，是因为我们定义了变量及使用了循环结构。其中循环结构就是控制结构的一种。

变量为啥能定义，也是因为程序语言定义了数据类型。绝大部分程序都用来处理数据，处理数据时，处理的对象就是已结构化的多种数据类型的组合体，这个组合体一般称为数据结构。所以程序语言的三大基本知识点分别是数据类型、控制结构、数据结构。

问：老师，我想问个问题，我刚进来这个群，不知道怎样才能更好地在这个群里适应（即我首先要干些什么，后干些什么，没头绪）

浩为：先看必读，再看看共享里以 0 开头的文件

问：我想在这个群里学些东西，明年就要毕业啦

浩为：下载书，按照书的说明，作为一个严格的执行者，估计你很快就能掌握 Java。不要去像前面那些兄弟一样沉迷于代码，那样只会浪费自己的时间。先从思想入手，就会很快明白为啥要那么写代码，而不是看了一堆代码，死记了很多代码，却不知道为什么。

对于乐意于死记代码的人我也难得再说什么，人各有所长吧，对于新手，我还是建议先从思想入手。有了思想做构架师很容易，如果只看代码，永远也做不了构架师。记住，编程的核心是思想，代码只是项目开发中很少的一部分，不要捡了芝麻，丢了西瓜。

通过认真理解前面内容及结合自己的学习过程，尤其是反复看这段话“我们先从最简单的代码入手，输入一个字符串‘Hello World’，而字符串就是一种数据类型。再输出多个。为啥能输出多个，是因为我们定义了变量及使用了循环结构。其中循环结构就是控制结构的一种。”思想结合几句简单的代码就把编程语言的核心都阐明了，可是以前看了许多代码都不知为什么。

我终于从代码的误区里走出来了，在此我更建议大家去看看《人生需要引导》。

学习小组能给大学生带来什么

来自: <http://zyt.howwe.net/me.php?294>

许童童整理，浩为喻桃阳修改

“对话浩为”中提到的问题归纳如下：（两人分别为浩为、许童童，及相应评语）

1、调查对两期【浩为资源堂月刊】的看法

浩为：我做的两期月刊看过多少，分别有什么感觉

许童童：文章都写得很好，但我觉得“浩为资源堂”没怎么推广，所以有些人看着不着，或者是有了之后也不会看

//许童童评（以下简称为“评”）：我觉得就算手中有这两期刊刊的也未必会去看，因为浩为现在不是那么出名，有些人仅看中名气…我觉得这是推广的事！

浩为：这是接下来要做的事，所以先调查一下

许童童：可以将 IT 多一点吗？感觉“人生”内容稍微多了些

浩为：第二期调整技术、人生分别为 16: 10，一共 26 篇，以后将控制在 2: 1。

浩为：给你的印象是什么

许童童：如果我不了解浩为的话，我可能会认为浩为只是个指导人生的

//评：我感觉在校大学生更关心如何学习技术，因为我身边很多同学，现在还处于没入门的状态，有些人现在都打算转行干其他了，大三还不了解一门语言！虽然大学开课了，但那些只是徒有其表，如果浩为期刊能多出点技术性的文章，在校大学生会很愿意去看！

许童童：浩为资源堂定位为在校大学生，是么？

浩为：对，宣传技术，其实更为以后的人生引导做长期规划

许童童：额，如果是我，我更关心技术。其实大学生懂很多人生方面的知识，但我感觉主要是实施起来不行，执行力不够

浩为：但是从我的经历看，全搞技术，进步会慢一点

许童童：可以往训练执行力方面加强

//评：说句实话，大学生也知道要学习技能，可惜大部分人受环境影响，整个寝室都在玩魔兽世界，却只有你在学习，即只有少数人选择去上自习，大部分人还是在跟同寝室的人一起玩！而且我觉得大学生的执行力不够！就拿我来说，我的执行力就不强，这亟待提高！因为我做什么事都是拖拖拉拉，本来想好的计划，基本上就没实施几个，实施了的也没按时完成！这说明我还不够自觉，我想有些人还是跟我一样！所以怎么去提高自己的执行力是我要思考的！因此我希望浩为能在这方面做些指引！

浩为：浩为资源堂接下来会组织学习小组，一起提高，其实也是一种促进，你觉得如何

许童童：这个可以！组怎么分？毕竟各自的水平不一样

浩为：所以要组织实地学习小组，实地加网络，网络安排学习内容，实地学习，遇到问题一起讨论、或网上答疑

许童童：额，不是那种兴趣小组，因为那样有些人会没有责任感

浩为：所以要加强责任感等的培训，有点类似职场培训，其实在《喻世名言新篇-人生需要引导》里提出了一套完整的引导学体系

许童童：哦，这个好。可学习小组是从哪些人里选？是群里还是 csdn？

浩为：浩为资源堂的人都可参加学习小组

许童童：但很少人会探究引导学体系的本质，我担心学习小组有些人会选择中途放弃！

浩为：正因为很少人会究其本质，所以很多人迷茫，不知道自己真正要做什么，或者没有毅力持续去做。

浩为：浩为资源堂的高校联盟其实就是为了学习小组准备的，同一高校的人更容易开始活动及交流

许童童：哦，那就是说同一个高校的人在一起或者同一个地区的人，但浩为资源堂在同一个高校的同学毕竟还是少数，我们学校在浩为资源堂就一个！

浩为：浩为资源堂接下来会将开始一系列的推广活动，目前正在谈一些合作

许童童：建议学习小组的任务由浩为高层来安排，监督也由高层来安排！而不是那种自觉型的！你看呢？

//评：很多人都会去选择中途放弃，一是因为自己毅力不够，还有是不知道自己想要什么。这也可以说是迷茫！怎样摆脱迷茫？认真去想自己想要什么？怎么做？这些做好之后再想我现在能做什么？然后从现在开始，一步一步朝着自己的方向驶去！如果中途想放弃了，就拿目标激励自己！

浩为的学习小组对在校大学生来说是很大的福音，因为大学正缺这种氛围，大学里都会有项目组，我们学校也有，我觉得项目组里的人只想自己如何提高，而不能相互提高，他们强调的是自身的强大！但浩为应该会与其不同，浩为是家公司，应该会按照企业的标准来要求学习小组的每一个成员！相信这些都是非常有用的！

2、浩为资源堂能得到什么？

浩为：你觉得在浩为资源堂里得到了什么？

许童童：我以前很少写文档，但现在我会去，而且愿意去，这是一个进步

浩为：这是一个很大的进步

许童童：以前觉得写文档只是浪费时间

浩为：26 号聚会就有一公司老总对我说，公司招人难，要求能写文档、会做活动策划、易与人相处

许童童：我同学身边没有几个愿意去写文档，估计也跟以前的我一样

浩为：写文档是一点，其他的呢

许童童：我在工作室一直强调经验分享，我觉得这点不错

浩为：交流更能促进提高

许童童：我在浩为资源堂里，还有认识了一批志同道合的人

浩为：其实这也属于资源整合

许童童：这也可以说是收获，虽然没见过面，但感觉很亲切

浩为：总的说来，这个社会还是好人多，浩为资源堂将打造一个良好的氛围

许童童：挺不错，这样学习起来带劲

//评：写文档确实能锻炼人，不要认为那是浪费时间的表现，其实在写文档的过程中你将会重新思考，而且自己的写作水平也能得到提高！所以我希望大家看完这篇文章后试着去写文档，然后养成这个习惯！

浩为在这强调浩为资源堂能给你带来什么，还有就是资源整合。真的！浩为会让你思考怎么去度过大学生活，能让你有一个明确的方向！我记得在跟我一高中同学（现在也大三）聊到大学的老师时，他说让他记忆深刻的老师有四种：1 是那种人贼好一到考试就给你题，而且上课从不点名；2 是贼 BT，RP 贼恶劣的那种，平时把学生骂得一无是处，还滥用手中那点权利折磨我们；3 是在上课的时候会跟我们唠唠家常，平时上课乐乐呵呵的！讲课有风趣的，讲的内容也爱听；4 是能给我们指条明路，把他们的经验说出来给大家听，让我们少走弯路！由于各种原因符合 3、4 的老师毕竟是少数！我觉得浩为就属于大学生喜欢的！如果学校按照企业的标准来教我们的话，我相信就不再会有学生毕业后抱怨找不着工作，企业也不会说找不找合适的人，也不用花时间去培训刚毕业的大学生。可惜学校没有！

学以致用：高为明对话浩为及感想

来自：<http://zyt.howwe.net/ok.php?319>

对话内容整理如下(两人为浩为喻桃阳、高为明)：

浩为：高为明，应该看过我和许童童交流的那篇文章“学习小组能给大学生带来什么”了吧。有什么印象？

高为明：看过了，浩为的理念非常好，但是感觉在推广上还需要做一些努力。

还有就是体会到“学以致用”的内涵了，以前学东西总是闷着头学，效果很不好，学完不用就忘了。以“用”为目的来学东西，在学之前先问自己要做个啥样的实例出来？把理论结合到实例中，这样学记得牢，事半功倍。

浩为：技术这块学得怎么样？

高为明：jsp 学的还可以，但在学校学习总有一种闭门造车的感觉。自己觉得学得不错，也许在实际中还差得很远。做过一些课程设计，SSH 框架基础学完了，但是没做过什么项目，整合还不太熟。

浩为：对感恩怎么理解？

高为明：感觉我们每个人要有一颗爱心，做好自己的事，然后尽力去帮助更多的人。能帮助别人是非常高兴的一件事。

浩为：毕业前对自己有什么安排？

高为明：都说框架很重要，一些工作的朋友也说企业用得比较多。在毕业之前要把框架学好，做一个像样的项目出来。然后再把 j2se 的基础巩固巩固。想着是明年 10 月份之前要找到一份工作，还有一年时间。前些天每天在看视频，终于看完了，寒假来了做一些东西练练手。

浩为：看视频收获如何，对自己的定位是做一个纯技术？

高为明：一些基础的东西都了解了，但是不太熟。感觉做技术是迈向职场的敲门砖，但不希望一辈子老做技术。

浩为：文档写作能力如何？

高为明：有时候脑子里有东西了挺能写，但有时候一点也写不出来。感觉自己有时比较焦躁，身边的同学也是，有时候什么事都不想做，很浮躁。写文档很不错，一来能练练自己的表达能力，二来写写文档，写写自己的点点滴滴和心得，总能从中得到点什么，不会再那么焦躁，生活、做事，一定要用心。

浩为：活动组织能力如何，活动是如何策划的？

高为明：大二那会我是在思想教宣部，有两个部长，我是其中一个。策划分给部员，每人写一份，然后交给我俩。组织过一次全校的教师技能大赛，还有几次系级的小比赛。大三学习紧了，活动参加的也就少了。

浩为：你怎么看浩为资源堂？对“浩为资源堂”推广规划有啥建议和感觉？

高为明：“浩为资源堂”是面向大学生的。很多大学生包括我身边的同学，他们老说迷

茫茫茫，想有个好的未来，但不知道应该怎么做，如何做？就是使劲不知道往哪使。现在感觉“浩为资源堂”思想上引导做的很好。技术方面的内容丰富一些就再好不过了。再多一些好的老师或有工作经验的人参与，老师和有工作经验的人发表的文章或写的一些心得往往极受欢迎。事实也确实如此。

浩为：浩为其实有很多系统及技术资料，现在还没放出来。不过我觉得零散的技术点，用搜索引擎一下就搜到了，所以更多的是告诉别人如何去使用搜索引擎。

高为明：授人以渔这种思想非常好，但目前最重要的是推广。

1、做一些对同学们非常感兴趣的东西，首先要把浩为推广出去。

当然做事情，要本着一颗为同学们服务的心，有核心的东西支撑肯定越飞越高。当然浩为也一直是这么做的。前段时间有个叫绝影的写了个《痴狂的程序员》，大家都爱看，他博客没什么东西，不过点击率却超过十万。好多博客点击率非常高，像 CSDN 有的博客上百万，也不知道怎么说，就是搜索引擎一搜索就能链到 CSDN 上。有个词叫什么 SEO 优化什么的，目的是推广，有什么方法都要拿过来用。

2、CSDN 学生大本营一出就火。当然其中有些老师在，受同学们欢迎。但更重要的是有 CSDN 这个大平台在后面做支撑，CSDN 大家都知道，大本营也就知道了。所以最重要是推广，通过某些方法，把浩为推广出去。但具体的方法就不知道怎么做了，希望对浩为有所帮助。

浩为：谢谢。浩为更多的是在大学实地推，而且不限于技术。因为纯技术的东西，面太窄，也没法长久。所以浩为准备找大学生，直接在大学里推学习小组，先做大影响。2010 年没意外的话，估计会投入几十万来做推广。<http://zyt.howwe.net/ok.php?284>这里面提到的只是一个轮廓。

高为明：嗯。以同学为轴心，先要做大影响。这点是最重要，但具体怎么做俺也不知道。不过有计划，然后去实践肯定会有效果。

一点点感触：跟浩为谈话最大的感触就是平易近人，不会让你感到压力。

1、关于学习方法：

以前学东西，只是一味的学。学 JAVA，看视频、买书，视频看完了、书看完了，也没什么感觉。总觉得比较空洞似的，没什么收获。还记得大概是 2009 年 5 月接触浩为，看过一篇文章：编程更需要引导：学以致用（<http://howwe.net/bk1/23.html>）有句话感触很深：浩为的观点是学以致用，掌握基本原理后，在实践中学习，而不是一开始就背代码。比如对 Java 基础的学习，在掌握三大基础的情况下，再去看 JDK 的 API 文档，搞清每个常用包的作用（即在你的脑海中建立一个模型，可用 FreeMind 自己做一个常用包的归纳图）就可以，用不着在每个类上花时间。学以致用，当你要用时，再去详细看一下 API 或 Google 一下，很快就掌握并能用了。

然后我开始摸索着学 JSP 和 SSH，不过这次学和以前不一样，先问问自己要做什么、怎么学，制定计划。构思一个实例，把理论结合到实例里，需要什么学什么，这样学就记得很牢，事半功倍。通过一个学期的学习，这学期 JSP 的课程设计得了 98 分。也许这和企业中的项目无法相提并论，但自我感觉这个学习过程特别充实。

2、关于写文档：

写文档是一件非常好而且有意义的事。我们要留心培养自己写文档的能力，最后把写文档当成一种习惯。

记得一个教授说过：你一个研究生连中文一级都不及格，你英文考六级干什么呢？看看

研究生写的论文，自己的民族文化都没有学好，天天考英语——打勾：托福打勾、GRE 打勾、英文考出很高的分。可哪个写的论文在我面前过得了关呢？根本就过不了关！这样培养出来的人能干什么？自己搞的专业一点都没学好！现在大学生毕业别的不说，文档写得好的真不多了，甚至还不如高中。从文档中能看出一个人的修为、涵养。多读多写，从中你会学到很多东西，做人，做事。

3、关于做人：

对于我这个年龄，也许根本没有资格谈做人做事这些大话题。不过还是想说，做任何事，一定要用心做。年轻人要勤思考，多总结，善提问。

4、关于推广：

对推广也没什么经验。不过相信浩为这样一个为大学生做实事的社区，一定会越飞越高。

浩为语：感谢高为明！文档等整理工作在职场是一个很重要的能力，工作中更多的是交流，可之后如何做，就需要整理，整理成档才能发现交流的不足，弥补不足之后才能更好地执行。否则就变成了神仙会，大家都吹吹牛，结果啥效果也没有。

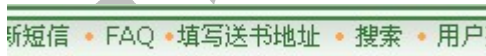
苦学勤练——技术高手的进阶之道


陈聚雄

工作之余，一有时间就抱起以前买的 Thinking In Java 再次读了起来。对于 Java 的学习，如果时间允许，建议每个人好好巩固巩固这些基础性的知识，特别是对于那些大学刚毕业，在学校并没有好好打牢基础的同学。如果能够充分利用自己空闲时间，把基础知识补好，对今后的发展将非常有利。

这个月，继续学习 PHP，边做一些小功能，边学习。学以致用用的观点，一直在我脑中不断加深，越来越感觉到自学能力的重要性。我觉得学以致用的核心是用以促学，在用中去学。

这段时间，在做申请送书功能，并把这个功能整合进论坛。对于我来说，功能实现不难，难的是如何与 phpbb 代码整合。所以开始读 phpbb 源码，不愧是牛人写出来的，读起来相当吃力，但能感受到代码中的艺术，因为看多了国人写的一些代码，再去看这些代码感觉是一种享受。里面的模版技术、缓存技术都是我以前没接触过的，其设计的思想和思路给了我不少启发。

在干活的过程中，还体会到一点：做事要细心，更要注意细节。就拿这次的送书页面来说，，大家看出这个图片上有什么不同了没？这是

我原来没有留意，后来发给浩为喻桃阳看时，他提醒了我，点少了个空格？，并说，在开始就应该注意这些东西，不要想以后再来修改。可见这是一个细心的问题。

前几日在 QQ 群上，看到几位高手聊天，说公司招人其实人才难求。其中有一位自主创业者奋斗七年，说如果是人才，可以出到 1.5w/人月。可见如果真是人才，就不用怕找不到工作，金子总是会发光的。他说他对 .net、c、c++、java、python、erlang、js、flex、gae、hadoop、amazon ec2、s3、tomcat、resin、ICE、grizzly、glassfish 等都研究过，

python 虚拟机、java 虚拟机的代码也仔细看过。

这几大高手强调，其实技术到一定程度，不用去管什么语言，解决问题一般会从行业的视角来思考，而不再是程序员的眼光。这些观点，与浩为喻桃阳的观点一致，可为什么到那种层次的人才会有那样的思想呢？如果我们从学习之初带着这种思想，去学习我们需要的知识，那在几年、十几年后，当我们达到了这个水平时，我们的观点是不是会更进一步呢？

在聊天过程中，他说到计算机这个行业，有时拼的是时间。无论你怎么聪明，总得花时间去实践以了解技术，即使把一些东西告诉你，当你没有一定基础，一般很难理解。如果你想学技术，并且准备从事这个行业，就得下苦功夫，学得扎实一点。

经过一段时间的了解，加上旁观高手论道，我觉得学技术的人有几个层次：

第一阶段、尽快成为高手。这个阶段不难，难的是需要努力加坚持。

第二阶段、实现自我价值。当你掌握了足够技术之后，做出很多产品来证明自己，发挥自己的价值。

第三阶段、站在行业角度思考问题，争取行业创新。

其实技术之外的很多东西更重要，好程序员一个月可以拿 3w。但是好职业经理人，一月可以拿 30w。这就是思想与行动的区别，这也是浩为教育提倡以思想促学习的功效。

旁观高手的聊天，虽然插不上话，但我从这些对话中，学到了许多课本上学不到的东西，更引发了自我思考，思索自己的未来，更加明确自己的方向，坚信自己朝着这条路走下去，定能快步走向胜利的彼岸。

第 1 篇 入门篇

入门，就是从不会到会的一个过程，搭建开发环境是入门的前提。

接着示范一个 Hello World 程序的开发。看清楚了没，程序开发就这样简单。

程序的开发离不开计算机语言，不论是何种语言：Java、C++、php，还是数据库代码，不外乎三种基本知识点：数据类型、数据结构、控制结构。

大部分程序代码就是这三种内容的组合，再根据软件工程的思想来实现功能。其中数据结构在 JavaBean 之后讲述。

讲完基础知识之后，接着演示一个简单数据库页面，使你对数据库开发有初步印象。但这只是最原始的方式，稍微高级一点的方式是 JSP+JavaBean 两层结构，再高级一点是 JSP+JavaBean+Servlet 三层结构 J2EE。

常用的 SSH (Spring+Struts+Hibernate) 是一种轻量级的 J2EE 解决方案，提出一堆的概念让人很迷茫，更是大大增加了学习的难度及时间，同时也没有解决 Java 在 Web 表现层 (UI，即用户界面) 开发的不足。Web 表现层变化频繁，单纯 J2EE 框架已无法满足需求，并且严重影响了开发及使用的效率。

RIA (富 Internet 应用程序) 技术，结合桌面应用程序反应快、交互性强的优点与 Web 应用程序的传播范围广及容易传播的特性，完全解决了 J2EE 在表现层的劣势。RIA 充分利用 Ajax 技术，将 Web 表现层完全或者绝大部分前推到客户端来开发，大大简化服务器端的 Java 开发。

浩为 HoCAS 是一个使用 RIA 的 SOA (Service-Oriented Architecture) 框架，表现层采用 JSP+jQuery 构架。通过自己定义的数据模型，动态生成界面 UI，将数据处理模块化，使用户的注意力集中在业务模型的建立上，详情请参考《从人生需要引导论 Java 快速开发》“第 26 章 基于 SOA 的 HOCAS”。注意：基础版 HoCAS 已取消 SOA。

HoCAS 拥有通用应用系统的管理模块，提供用户管理、权限管理、报表模块，通过先定义表、根据表配置页面，菜单管理器配置菜单来调用页面，随时拓展功能，从而实现项目需求。《从人生需要引导论 Java 快速开发》中“第 18 章 基于 HoCAS 的项目管理系统”显示了 HoCAS 的便利。

2009 年 4 月，浩为与高校合作，开发基于 HoCAS 的形式化建模工具，采用 UML 建立业务模型，根据业务模型直接生成业务系统。详情请参考《从人生需要引导论 Java 快速开发》“第 25 章 基于 HoCAS 的 UML 建模”。

电话外呼系统 HwCall 现已应用于多个地市气象局，作为标注 J2EE 范例，能让你快速掌握 Web 项目的开发，早日踏入成功之门。详情请参考《从人生需要引导论 Java 快速开发》“第 14 章 电话外呼系统”、“第 15 章 HwCALL 系统设计”及“第 16 章 HwCALL 编码”，本书第 16 章只介绍了基本实例。

第 1 章 搭建开发环境

搭建开发环境，是进行数据库开发的前提。通过本章的学习，能快速搭建一个开发环境。

由于开发环境的配置繁杂，初学者一时难以理解，浩为推出 HoCAT。HoCAT 基于 Tomcat，提供运行监控、自带 JRE、Windows 服务管理，绿色软件。

为了使用更加方便，浩为将本书使用的软件整理成压缩文件，放在互联网上随意下载。压缩文件 work.rar 包含 JRE、HoCAT、Eclipse、FreeMind 等软件及范例，近 70M，1 个文件或 7 个文件，其中 Eclipse 版本为 3.3 完整版，仅去掉 datatools（2009 年 4 月最新版 3.4 不支持 jQueryWTP 插件）。不用安装其他软件，仅需按说明执行一下脚本即可。

1.1 Java 运行环境 JRE

Java 程序的运行必须要安装 Java 运行环境，我们可以通过安装 JRE 或者 JDK 获得。所有 Java 代码首先被编译后缀为 .class 的类文件，然后类文件在虚拟机 JVM 上执行。

JRE（Java Runtime Environment，Java 运行环境），运行 Java 程序所需环境的集合，包含 JVM（Java Virtual Machine）标准实现及 Java 核心类库。

JDK（Java Development Kit），Java 开发工具包，里面包含了各种类库和工具，其中包含 JRE。

JVM是一个虚拟的计算机，能模拟各种计算机的功能，它有处理器、堆栈、寄存器等虚拟的硬件架构，及相应的指令系统。JVM屏蔽了与具体操作系统平台相关的信息，使得Java语言编译程序只需生成在JVM上运行的目标代码(字节码)，就可以在多种平台上平稳运行，而不需要修改，从而实现Java程序的跨平台。

JRE/JDK可以从<http://java.sun.com/javase/downloads/index.jsp>下载，如果我们只是应用而不是开发，只下载JRE就足够。

1.2 Web 服务器

Tomcat 是一个轻量级应用服务器——Web 服务器，是开发和调试 JSP 程序的首选。

Tomcat 是由 Apache 软件基金会下属的 Jakarta 项目开发的一个 Servlet 容器，按照 Sun 提供的技术规范，实现了对 Servlet 和 JavaServer Page (JSP) 的支持，提供的 Jasper 编译器将 JSP 编译成对应的 Servlet。

Tomcat 本身内含一个 HTTP 服务器，一般可作为单独的 Web 服务器。Tomcat 通常与 Apache 或者其它 Web 服务器一起工作。除了用于开发过程中的调试以及那些对速度和事务处理较小的用户外，将 Tomcat 单独作为 Web 服务器的用户在前几年不多。但随着版本的更新，正有越来越多的用户将其单独作为 Web 服务器，用以那些对速度和可靠性有较高要求的环境中。

Tomcat 使用 Java 开发，开放源码，可以运行在任何一个装有 JVM 的操作系统之上。

下载地址: <http://tomcat.apache.org/download-60.cgi>

小知识：什么是服务

Tomcat 是一个 Web 服务器，提供解析网页的服务。到底服务是什么呢，简单一点说，服务就是实现功能的程序，一般用于提供基于网络的应用。

服务的工作原理：监听某一端口，接受此端口的请求，返回相应的内容。如 Tomcat 监听 80（默认监听 8080），接受 URL 请求后，返回对应的内容。

常用服务：

80 Web 服务端口，访问时，URL 地址隐含了一个默认的端口 80，如果不为 80，必须在 URL 中加端口号。很多协议都有默认的端口，如 ftp 为 21、smtp 为 25。

139 为“NetBIOS Session Service”提供的，主要用于提供 Windows 文件和打印机共享以及 Unix 中的 Samba 服务。在 Windows 中要在局域网中进行文件的共享，必须使用该服务。

1433 数据库 SQL Server 的默认端口。

3306 数据库 MySQL 的默认端口。

9002 数据库 hsql 的默认端口

在复杂的企业应用中，可能会在同一个服务器上安装多个厂家的产品，而这些产品中就可能使用多个数据库，如 MySQL 数据库。但为了管理的方面，提供产品的厂家一般只使用自己的数据库，这样就必须在服务器上通过使用多个 MySQL 程序监听多个端口的方式来提供服务。

测试某个端口号（如 139）是否可用，可在命令行（DOS 窗口），输入“telnet 127.0.0.1 139”，如出现错误提示则此端口号不通，如图 1-1 所示。

```
C:\>telnet 127.0.0.1 139
正在连接到127.0.0.1...不能打开到主机的连接， 在端口 139：连接失败

C:\>netstat -an

Active Connections

Proto Local Address           Foreign Address         State
TCP   0.0.0.0:135              0.0.0.0:0               LISTENING
TCP   0.0.0.0:445              0.0.0.0:0               LISTENING
TCP   0.0.0.0:1025             0.0.0.0:0               LISTENING
TCP   0.0.0.0:1026             0.0.0.0:0               LISTENING
TCP   58                      :139                    0.0.0.0:0               LISTENING
TCP   58                      :1547                   64.4.34.130:1863        ESTABLISHED
TCP   58                      :1570                   218.189.38.21:80        CLOSE_WAIT
```

(图 1-1 服务及端口)

如果要查看所有端口，在命令行（DOS 窗口），输入“netstat -an”，结果如图 1-1 所示。

其中 State 为 LISTENING，表示此端口正在监听中，即可提供服务；ESTABLISHED 表示两个 IP 地址之间已经建立连接。

木马程序的原理：先在电脑上用木马程序监听一个端口，即此端口的状态为 LISTENING，

其他电脑通过网络访问此端口，建立一个连接（其状态为 ESTABLISHED），就可任意操作被控制者的电脑。另一种方式是定时从某电脑上获取指令，在电脑后台中执行指令，譬如定时访问某个页面（简称刷票）。

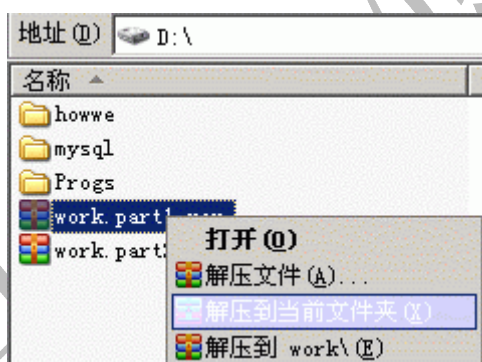
为了电脑的安全，建议将不用的端口全部关闭，或使用防火墙关闭对端口的访问。

1.3 Java 开发整合包使用说明

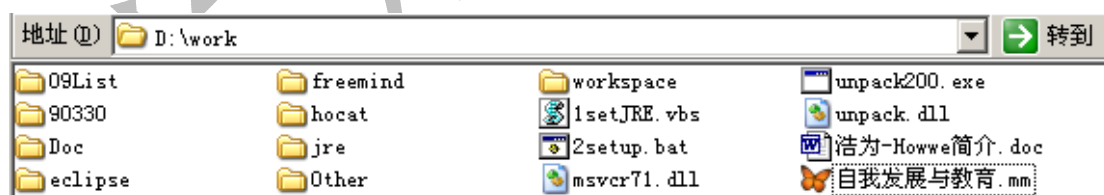
由于 Jre、Tomcat 等软件需要安装及配置，初学者一时难以理解，浩为曾推出 HoCAT。HoCAT 基于 Tomcat，提供运行监控、自带 JRE、Windows 服务管理，绿色软件。

搭建Java开发环境，是进行数据库开发的前提。由于开发环境的配置繁杂，将给初学者带来麻烦，浩为将本书使用的软件整理成压缩文件，放在互联网上随意下载。压缩文件 work.rar 包含 JRE、HoCAT、Eclipse、FreeMind 等软件及范例，近 70M，1 个文件或 7 个文件，其中 Eclipse 版本为 3.3 完整版，仅去掉 datatools（2009 年 4 月最新版 3.4 不支持 jQueryWTP 插件）。仅需按说明执行一下脚本即可。下载 <http://howwe.net/tools>。

文件全部下载后，请放在 d:\根目录下，文件的解压如图 1-2 所示，图 1-3 为解压后目录 D:\work 下的部分内容，如有不同，请调整目录。



(图 1-2 work 解压)



(图 1-3 解压后的目录 d:\work)

安装步骤：（D:\work 目录结构后面详细介绍）

提示：如果已经安装 JDK，并且要使用已经安装好的 JDK，请删除目录 jre、自己配置环境变量、及去掉文件 2setup.bat 中对 jar 及 jnlp 的文件关联部分。对于初学者，请严格按照书介绍的步骤，并且工作目录必须为 D:\work。

1、配置环境变量，有两种方式：自动配置、手工配置，任选一种即可。此步将配置两个环境变量：JAVA_HOME、path。

自动配置：双击 1setJRE.vbs 运行脚本，提示 set ok 即可。

手工配置：从资源管理器 => 我的电脑，单击右键，选择“属性” => 高级 => 环境变

量。如图 1-4 所示的窗口中，先修改 path，在变量值中添加 ;d:\work\jre\bin; ；再“新建”，变量名为 JAVA_HOME 、变量值为 D:\work\jre 。

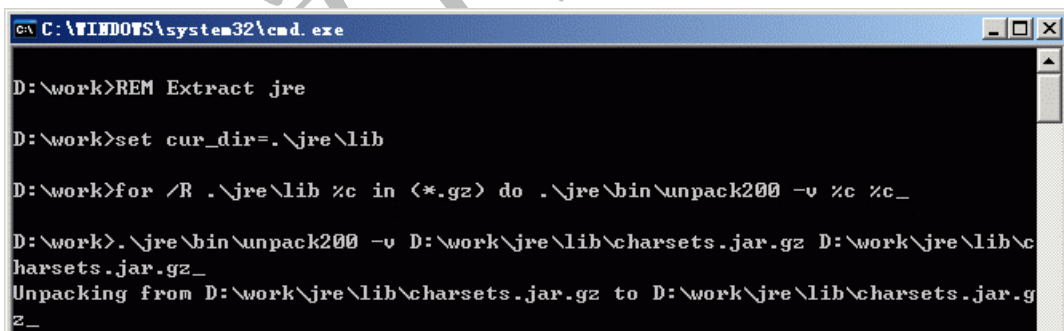
两值均可根据自己的配置而修改，也可以修改文件 1setJRE.vbs 的相应路径后运行。



(图 1-4 添加环境变量)

2、先运行（鼠标双击） D:\work\2setup.bat （如没有此文件，则为 2setup ，下同）

如图 1-5 所示，开始解压被压缩的运行程序，解压完成，此窗口将自动关闭，1setJRE.vbs 和 2setup.bat 也被删除。

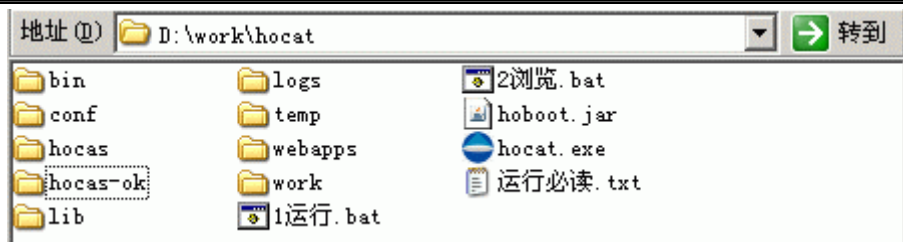


(图 1-5 文件解压)

运行 Web 服务器

如图 1-6 所示，鼠标双击 D:\work\hocat\1 运行.bat 或 hocat.exe，启动 Web 服务器 Tomcat。

也可以运行 D:\work 下的快捷键：快捷方式 到 hocat.exe。

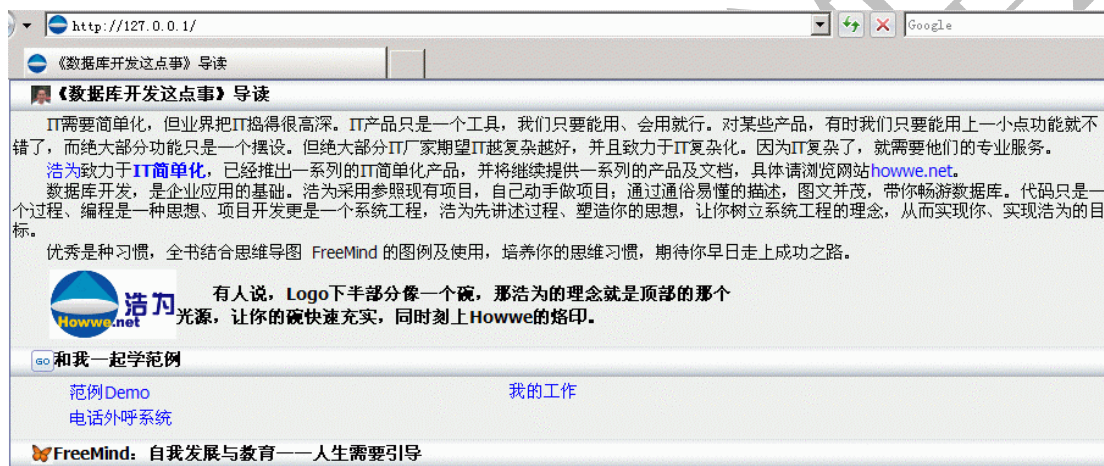


(图 1-6 HoCAT 目录结构)

其中 hocas-ok 目录为本书的示范项目，hocas 为你将完成的项目。

浏览网页

运行 D:\work\hocat\2 浏览.bat，将启用浏览器查看网页 <http://127.0.0.1>，可显示如图 1-7 所示的简易页面，不过在本书发行时具体的页面内容可能会有所变化，但“《数据库开发这点事》导读”总是第一行，否则，就是你的 80 端口已被占用。



(图 1-7 HoCAT 首页)

也可以直接在 Eclipse 中运行 Web 服务器及浏览网页，具体请浏览下一节。

注意：如果 80 端口已被占用，请修改，在文件 D:\work\hocat\conf\server.xml 中查找 `<Connector port="80">`，将 80 改成你需要的端口，如 89，修改后请同时修改“2 浏览.bat”文件中的 80。

D:\work 为 HoCAT 上一级目录，可作为工作目录的范例。



(图 1-8 Work 目录结构)

图 1-8 使用 FreeMind 生成，这是 FreeMind 的一项功能：将目录导入。

使用这一功能，可以更直观地显示目录及文件的结构，同时可以对目录或文件添加说明，如图中所示的“文件说明范例”。如要打开文件或目录，移动鼠标至文件或目录前的图标，待鼠标图标变为手，即可打开。

注意：尽量不要在 FreeMind 文件中删除文件或目录，因为有时可能会误删除，即在不知不觉中文件可能没了。如想修改，最好将 mm 文件保存在另外的目录，即 mm 文件不能和导入的目录在同一目录或父目录中。

目录及文件说明：

- 09List 09 年工作目录，存放备份的日工作内容，可用于为事后整理及分析总结。
- 90330 2009 年 3 月 30 日的工作目录，记录当天的工作、相关的文档。需定时归档。
- eclipse java 编辑工具，下节详细介绍。
- Doc 存放常用的文档。
- hocat Web 服务器，基于 Tomcat，自带 JRE。
- freemind 思维导图（MindMap）软件，一种开源 Java 软件，可以帮助你整理思路，英国人托尼·巴赞创造的一种笔记方法，它以直观形象的图形建立起各个概念之间的联系。
- Other 存放其他文件
- jre Java 运行环境
- workspace eclipse 的工作目录
- 自我发展与教育.mm freemind 格式的文件，揭示定位对个人发展的重要性
- 浩为-Howwe 简介.doc 浩为公司简介。

在日常工作中使用如此结构的目录，将有助于工作的顺利开展，形成一种良好的工作习惯。

比如日常文件的管理，养成按日期存放及整理的习惯。以后要找文件时，不会找不着。

1.4 Eclipse 使用简介

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。用户一般将 Eclipse 当作 Java IDE 来使用，用于编辑 Java 代码（Jsp、Jsf 也可属于 Java 代码），也可以编辑 php、C 等语言。

下载地址：<http://www.eclipse.org/downloads/>

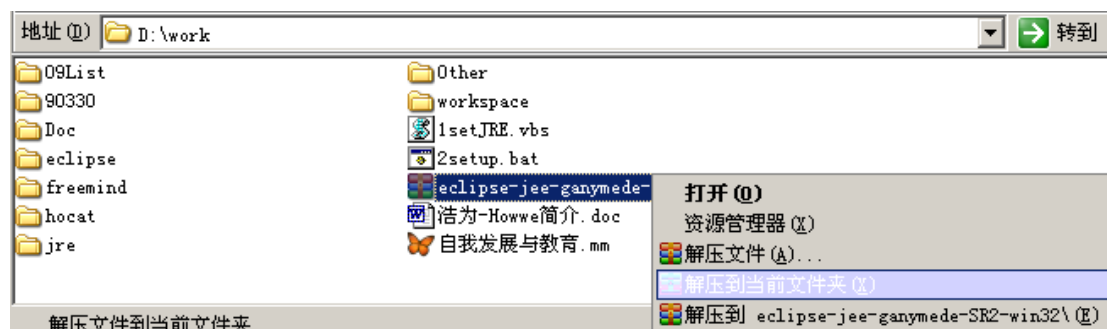
如果你想自己配置 Eclipse，可下载 Eclipse，步骤如下。



1、选择 **Eclipse IDE for Java EE Developers**，2009 年 4 月当前版本为 3.4.2（注意，此版本不支持 jQuery 的整合插件 jqueryWTP），下载文件为 eclipse-jee-ganymede-SR2-win32.zip。

2、下载后，将文件 eclipse-jee-ganymede-SR2-win32.zip 复制到目录 d:\work 下，单击右键，如图 1-9 所示，选择“解压到当前文件夹”，解压后的文件全部在目录

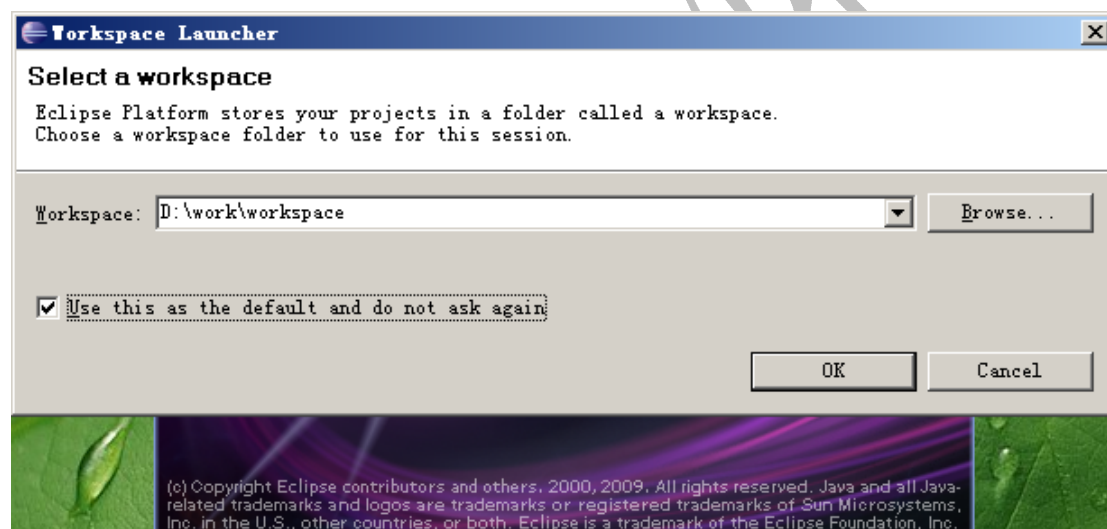
d:\work\eclipse 下。



(图 1-9 解压 Eclipse)

注意，已经存在 eclipse 目录，解压时请选择“全部覆盖”。目录下的文件将在后面用到。

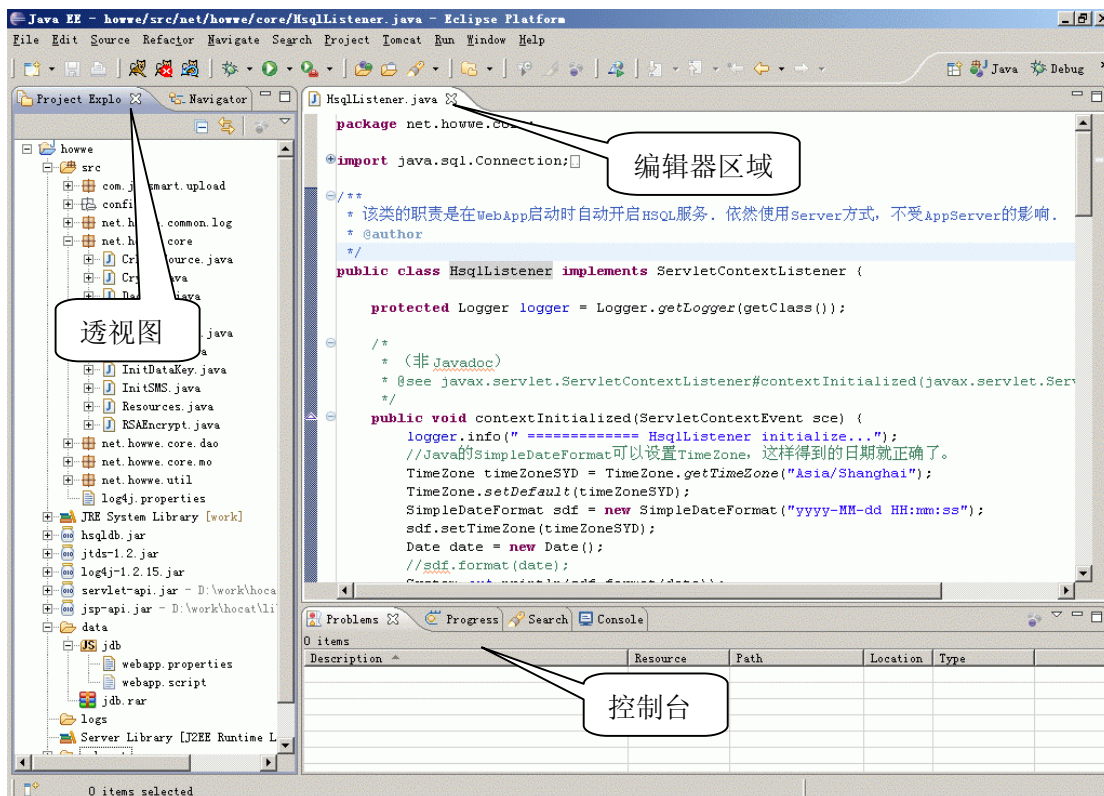
双击 d:\work\eclipse 目录下 eclipse.exe，也可以双击 d:\work 下的“快捷方式 到 eclipse.exe”。由于 Eclipse 将加载很多插件，如果运行的电脑配置不高，可能需加载几分钟才能完成。第一次运行 Eclipse 时，将出现如图 1-10 所示的“选择 Workspace”界面，请选择 D:\work\workspace，此目录下有 Eclipse 的配置文件及项目信息，便于初学者使用。



(图 1-10 第一次运行 Eclipse 选择 Workspace)

注意：1、使用整合包不会出现图 1-10 所示界面。2、不推荐使用 MyEclipse，此软件太占系统资源，在很多电脑操作不顺畅。

Eclipse 启动成功，界面如下：（注意，编辑器区域打开的文件与图可能不一致）



(图 1-11 Eclipse 工作台)

注意：如果 Project 视图区的项目 howwe-ok 显示为红叉上，表示此工程有错误，按 F5 刷新即可。

Eclipse 工作台由几个称为 **视图 (view)** 的窗格组成，比如左上角的 Project 视图。窗格的集合称为 **透视图 (perspective)**。

Project 视图 允许你创建、选择和删除项目。

右侧的窗格是**编辑器区域**，用于编辑文件，如图 1-11 所示的文件 HsqlListener.java。

右下角区域为**控制台**，控制台里面的部分称为**标签**，用于显示和项目相关的内容及加载的插件。常用的标签如下：

Problems: 项目编译后的提示信息，包括提示、错误等；

Progress: 工作提示窗口，如 Eclipse 对项目的编译、插件的更新等；

Search: 查询结果；

Console: Tomcat 运行日志。

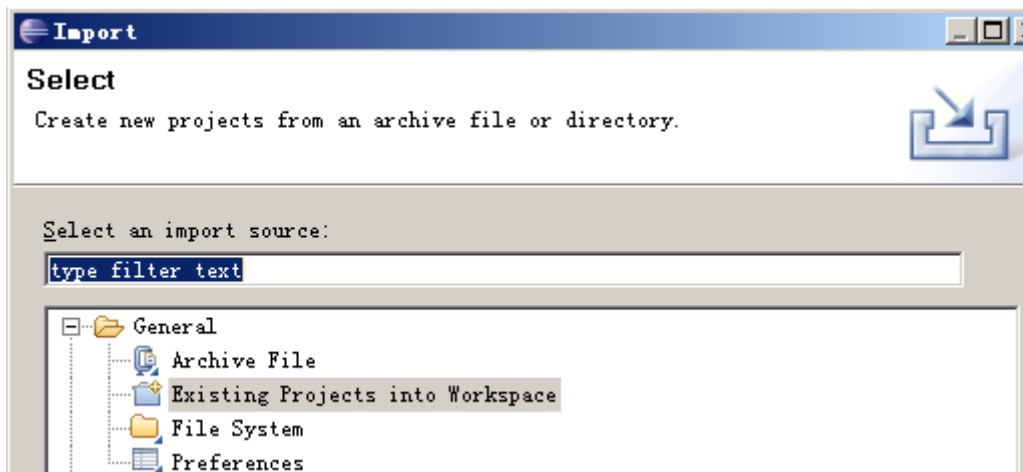
Connections: 数据库管理插件 SQLExplorer 的连接管理窗口，相关标签两个：Database Structure 和 Database Detail。详情请参考“第 9 章 SQLExplorer 使用简介及 SQL 简介”。

1.5 新建 Eclipse 项目

一、项目导入

操作步骤：**File => Import**（中文版对应为 **文件 => 导入**），在如图 1-12 所示的界面中选择 **General => Existing Projects into Workspace**（中文版对应为 **标准 => 现有项**）。

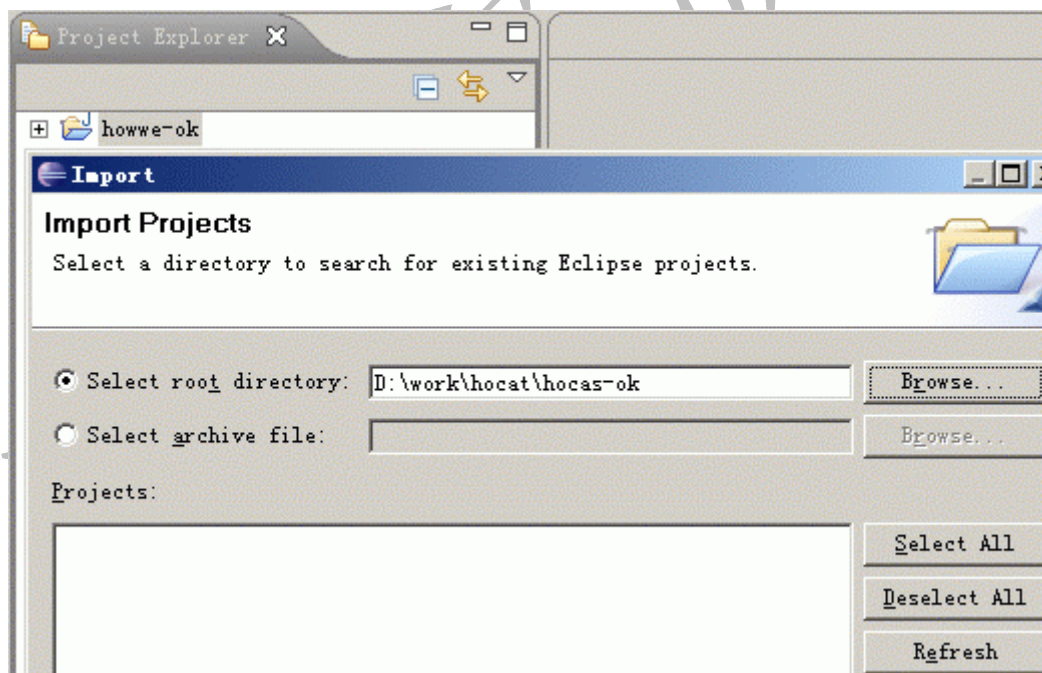
目导入工作空间)，点 Next (中文版对应为 下一步)。



(图 1-12 Eclipse 导入项目)

在图 1-13 中，点按钮 **Browse** (中文版对应为 **浏览**)，再选择项目目录 D:\work\hocat\hocas-ok，因为 howwe-ok 已经存在，故不能再导入。如果要导入的项目没有加载过，选中要加载的项目后点按钮 **Finish** (中文版对应为 **完成**) 即可。

注意：如果项目 howwe-ok 已经存在，就不能再导入。项目导入时，先将检查已存在的项目名，如果已经存在，就不会在如图 1-13 所示的项目列表 (Projects:下面) 列出。



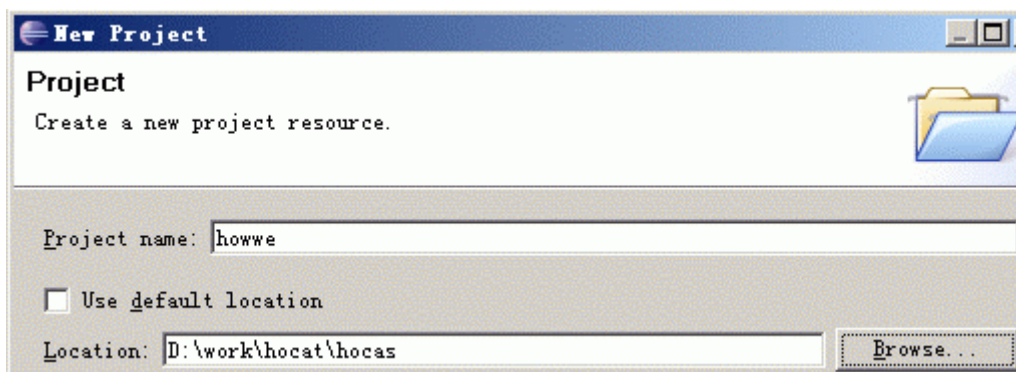
(图 1-13 Eclipse 导入项目)

二、新建项目

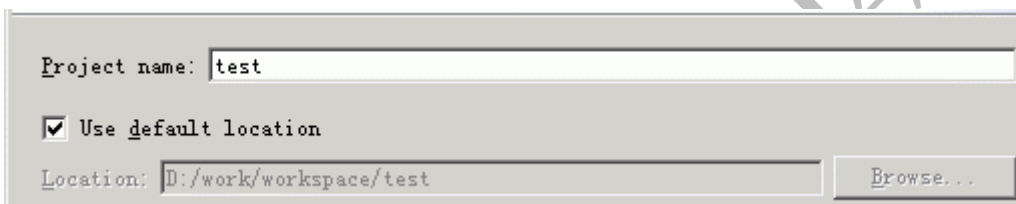
操作步骤: **File** => **New** => **Project** (中文版对应为 **文件** => **新建** => **项目**)，在出现的窗口中选择 **General** => **Project**，如图 1-14 所示的窗口中，在项目名 Project name 输入 howwe，取消 Use default location 前面的选中，点 **Browse** 选择目录 D:\work\hocat\hocas 后 **Next**，再 **Finish**。

注意，如果选择 Use default location，将在工作空间 workspace 下自动创建一个以

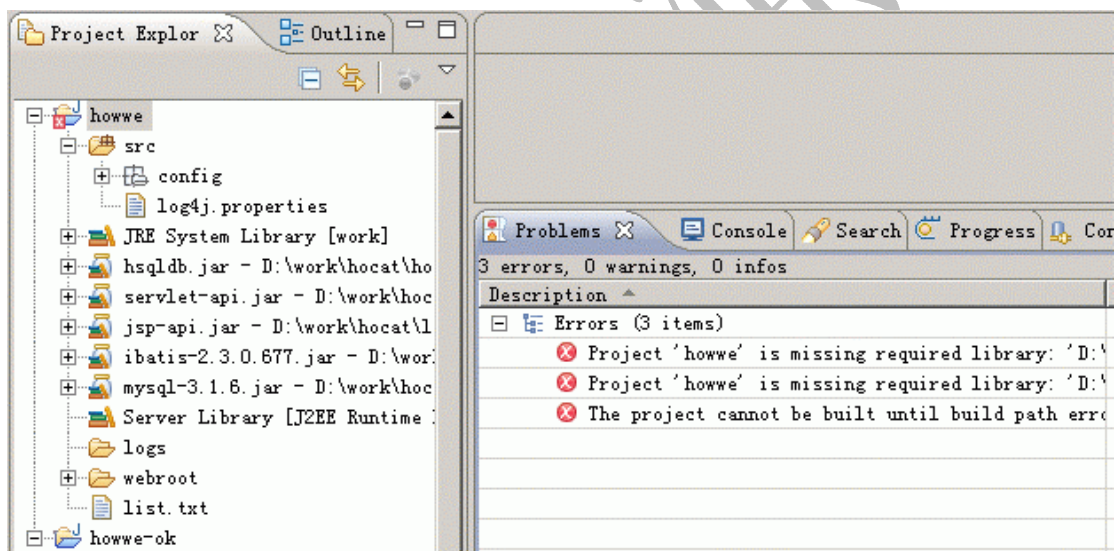
项目为名称的目录，如图 1-15 所示。



(图 1-14 Eclipse 新建项目)

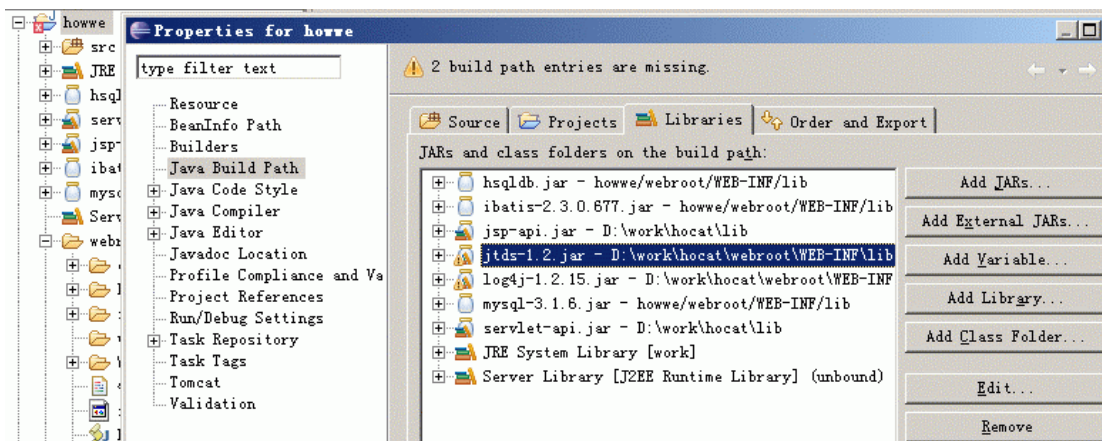


(图 1-15 Eclipse 新建项目- Use default location)



(图 1-16 Eclipse 新建项目 howwe)

新建的项目 howwe 如图 1-16 所示，项目名 howwe 显示为红叉，表示有错误，看右下角控制台里 Problems 的提示，显示缺少库文件 jar。在项目 howwe 处单击右键，选择 Properties，出现如图 1-17 所示的窗口。



(图 1-17 修改库文件位置)

修改路径，先选择 Java Build Path => Libraries => 第 4 项 jtds-1.2.jar，有两种方式：

- 1、选中后，单击按钮 **Edit** ；
- 2、直接双击文件 jtds-1.2.jar。

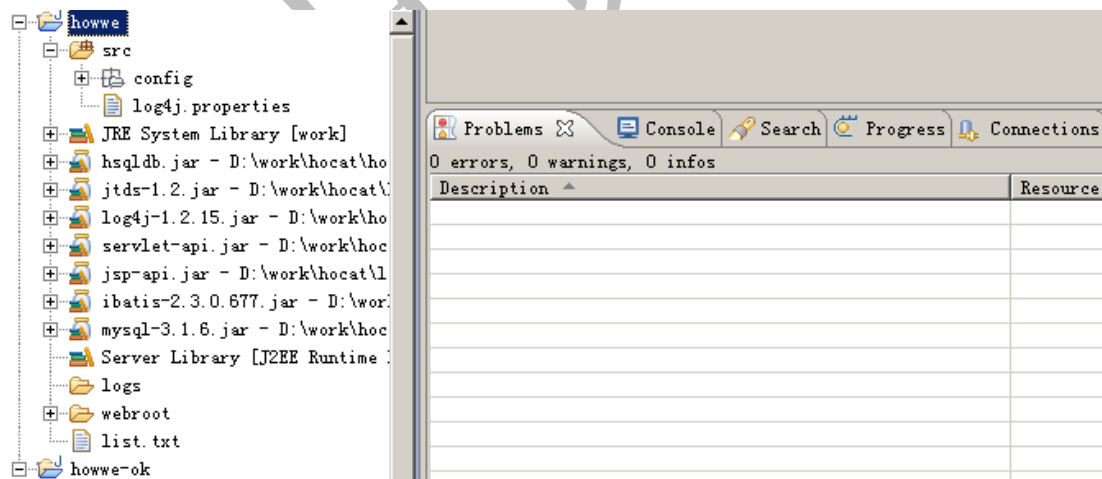
再选择目录 D:\work\hocat\hocas\webroot\WEB-INF\lib 下的 jtds-1.2.jar。图标左下角有提示图标的项表示有问题，请用同样的方法修改。

修改完 2 个 jar 的位置后，点 **ok** 后返回，如图 1-18 所示，已修正项目错误。

常用按钮说明：

Add JARs 添加本项目下的 Jar 文件（文件位置：WEB-INF\lib），也可以添加相关项目及正打开项目中的 Jar；

Add External JARs 添加指定目录下的 Jar。

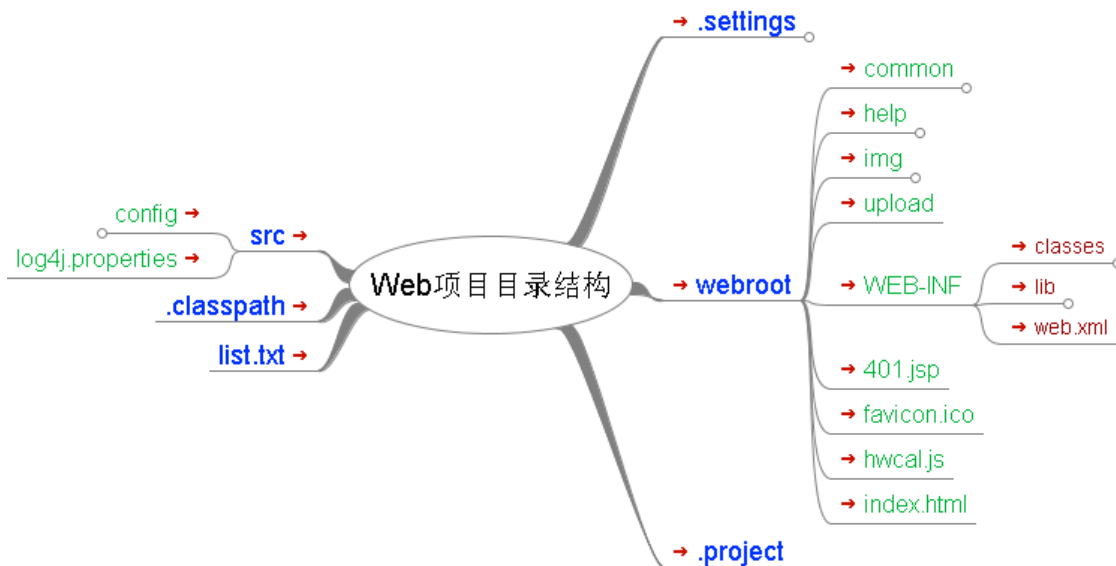


(图 1-18 项目错误已修正)

三、新建 Web 项目目录结构说明

新建的项目请按一定格式来新建目录和存放文件。对于初学者来说，新建时直接选择现有项目的目录即可（新建前请将一成熟项目 copy 过来）。

目录结构如图 1-19 所示。



(图 1-19 项目目录结构)

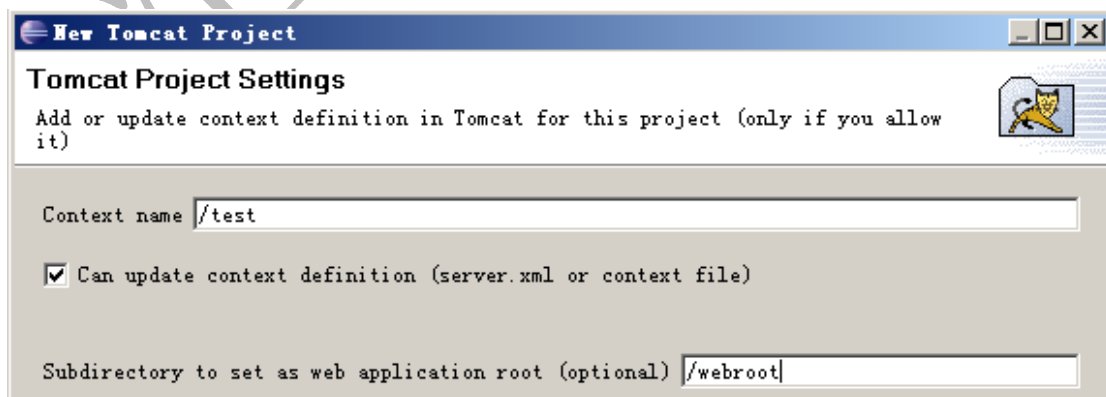
目录及文件说明：

- 目录 src 存在 Java 源文件：包、类、配置文件也可放，如图 1-21 中的目录 config 存在的是数据库配置文件及参数。
- list.txt 项目变更基本说明。
- 目录 webroot Web 项目发布目录。其中 WEB-INF 为固定目录，下面有 classe 存放编译后的 class 文件、lib 存放项目需要用的 jar 文件（也可以将项目生成的 class 和 jsp 编译后生成的 class 文件打包成 jar）。
- 目录 img 存放图片及 css、js 等文件。

如果要建立一个全新的项目，请参考“1.6 新建 Tomcat 项目”。

1.6 新建 Tomcat 项目

操作如下：File => New => Project（中文版对应为 文件 => 新建 => 项目），在出现的窗口中选择 Java => Tomcat Project，在项目名 Project name 输入 test，再 Next，在图 1-21 所示的窗口中设置 Web 应用的根目录为 webroot，再 Finish。



(图 1-21 设置 web root)

注意：如果你自己配置 Eclipse，必须加入 Tomcat 插件才可。

新建完成后，插件自动在 Tomcat 相关目录 D:\work\hocat\conf\Catalina\localhost 下添加项目配置文件 test.xml，如图 1-22 所示：

地址 (D) D:\work\hocat\conf\Catalina\localhost		
名称	大小	类型
bak		文件夹
demo.xml	1 KB	XML 文档
me.xml	1 KB	XML 文档
test.xml	1 KB	XML 文档

(图 1-22 已添加项目配置文件)

test.xml 内容如下：

```
<Context path="/test" reloadable="true" docBase="D:\work\workspace\test\webroot"
workDir="D:\work\workspace\test\work" />
```

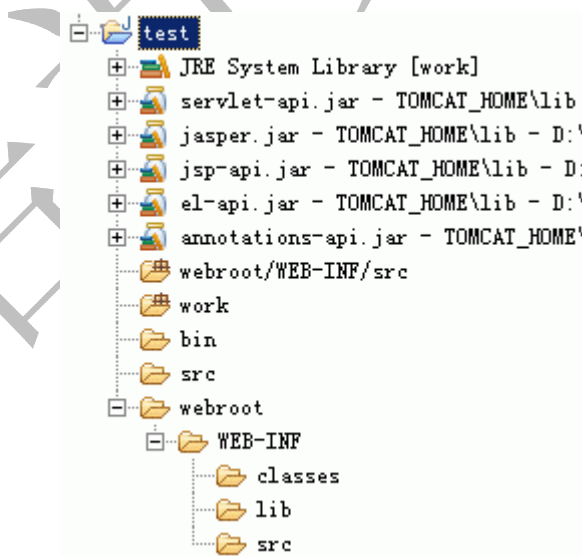
参数说明：

path="/test" 在浏览器中可通过test来访问，完整地址<http://127.0.0.1/test>
 reloadable="true" 当 test 项目中 java 代码有修改时，将自动重新加载
 docBase="D:\work\workspace\test\webroot" 项目文件地址
 workDir="D:\work\workspace\test\work" 项目工作目录，存放 jsp 编译后的文件

具体请参考《从人生需要引导论 Java 快速开发》“第 19 章 Web 应用工作原理”。

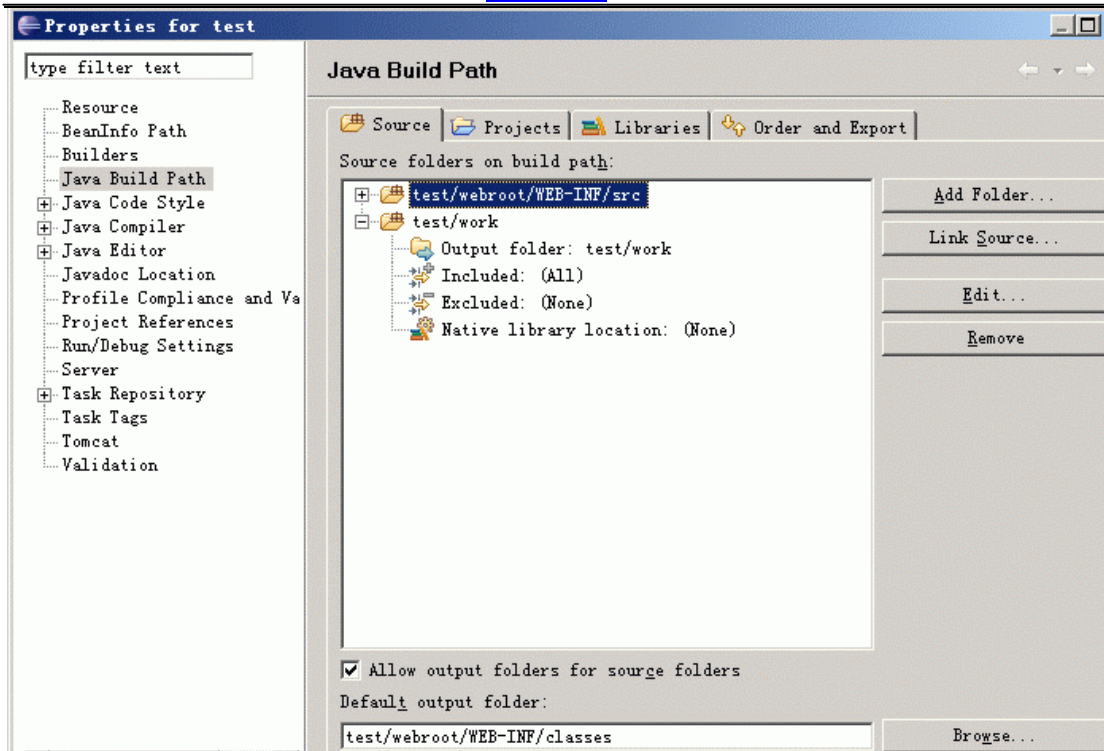
插件同时生成如图 1-23 所示的项目目录：

注意：如果你不想在项目发布时，将项目的源代码也发布出去，建议修改一下目录。修改步骤如下：

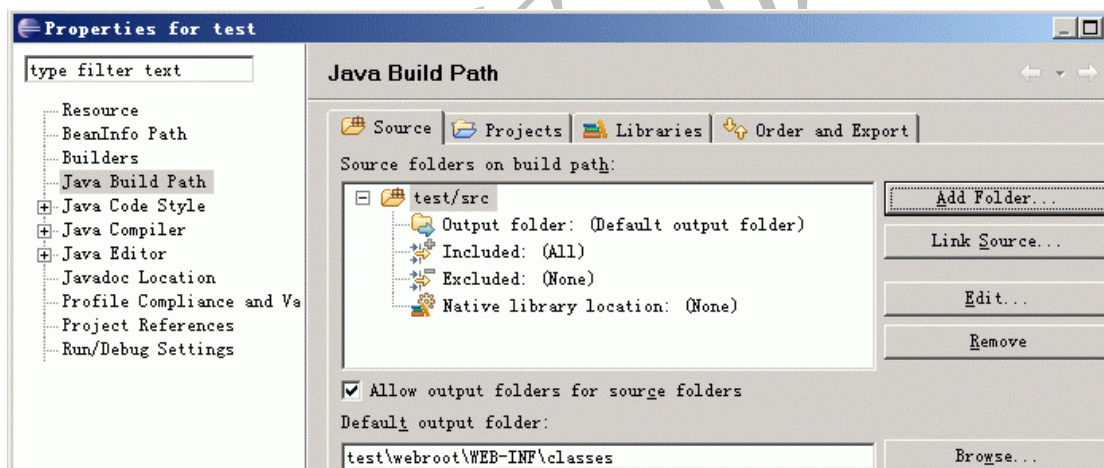


(图 1-23 项目目录)

1、在项目 test 单击右键，选择 **Properties**，出现如图 1-24 所示的窗口，先依次选中 **Source** 中的两项，点击按钮 **Remove** 去掉选中的内容。再 **Add Folder** 选择目录 **src** 后 **ok**。如图 1-25 所示。

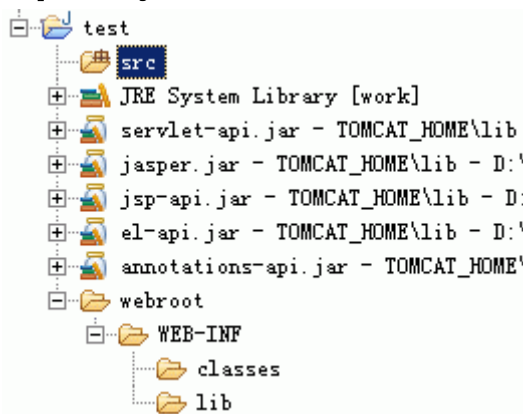


(图 1-24 调整项目目录)



(图 1-25 添加 src 目录)

2、删除目录 bin、work 及 webroot\WEB-INF 下的 src，然后在项目 test 单击右键，选择 **Close Project**，再 **Open Project**，则调整后的项目目录如图 1-26 所示。



(图 1-26 调整后的项目目录)

最后将项目中需要用到 jar 文件放在 webroot\WEB-INF\lib 下，在项目 test 单击右键，选择 **Properties**，在 **Libraries** 中用 **Add JARs** 将 jar 加入项目即可。

注意：建议将需要用到 jar 文件全放在 webroot\WEB-INF\lib 下。但在 Tomcat 下，不能将 jsp-api.jar 和 servlet-api.jar 直接放在 lib 目录下，否则 Tomcat 启动时报错，可放入新建的目录 ext 下。

第 2 章 激发你的思维潜能

思维导图是用图形化的方式展现想法和概念，作为一个可视化思考工具，它有助于信息的结构化，便于更好的分析、理解、综合、回忆以及产生新的想法。正像所有的好点子一样，它的能力就在于它的简单，其中 FreeMind 是思维导图的典范。FreeMind 的使用特别简单，以致我都不想太多的篇章来介绍其使用。

形象思维是运用头脑中积累起来的表象进行的思维。表象是我们以前知觉过的，而在头脑中再现的那些对象的映象。即现实中的苹果或描叙的苹果是对象，通过形象思维处理后，在我们的大脑中形成苹果的映象。

随着思维导图使用的增多，思考问题的条理性大有提高，在无形中就提高了你的思考能力，更激发了你的思维潜能。

使用 FreeMind 做做日程管理，事前计划及安排日程、工作，事后及时跟踪。将培养自己持续努力的习惯，永葆干劲十足。

人有偷懒的天性，面对漫长的人生之路，很多人的失败就在于不能持续以恒。人必须认清自己，方能激发自己的潜力，使用 FreeMind 软件或思维导图的思想来分析问题及实现“日程管理”，坚持一段时间，就能发现自己的思维习惯及日程等大有改善。有人说，优秀是种习惯。人生需要引导结合思维导图 FreeMind 等的使用，辅以日程事务管理，培养你的思维习惯，期待你早日走上成功之路。

2.1 我的思维导图使用历程

几年前我就知道思维导图，曾使用 FreeMind 做过一些分析，但一直没有细究，更没有深入使用。2008 年年底，我请一朋友帮我对公司的定位做分析，我先列出四大方向，他接着分析各方向的优劣，讨论后确定公司以培训为主，所以 2009 年转向 IT 培训。

在写书之前，做了一些产品，其中就包含 FreeMind 修改版。因为我觉得这个软件不错，值得推荐给别人，但还是没有上升到能提高思维能力的层次。

在做“HwCall 电话外呼系统-技术方案”时，使用 FreeMind 做了几个系统构架图，觉得这个软件很不错。于是在撰写《从 Java 快速开发谈人生需要引导》时开始使用 FreeMind，尤其是在分析“自我发展与教育的关系”（人生需要引导相关概念的第一版）时，突然意识到那朋友给我做公司定位分析时，其中就包含思维导图的思想。

这样就对思维导图的重要性有了更深的认识，于是在书中大量使用 FreeMind，同时也查了不少思维导图的资料，以致在该书中用一章来介绍思维导图。

在 QQ78928780 系列群中，特意要求群员使用 FreeMind 为自己做一个“近期学习安排”，即后面提到的“日程管理”，做过的人都觉得对他们很有帮助。

使用思维导图几个月，发觉自己思考问题的条理性大有提高，譬如在思考某论坛的改版时，我想到的是哪些人会来论坛，而论坛能为这些人提供哪些内容，列出了几大人群，觉得论坛从这个角度去考虑改版，即论坛的改版先得考虑定位。

暑假期间，我同时在安排人参与《软件工程术语中英对照案例解析教程》的编写，经常和我老师交流，我最关心的是教程的定位，即哪些人会来看这书。这样很快就抓住了问题的本质，从而将问题解决得更加完美。

人有偷懒的天性，面对漫长的人生之路，很多人的失败就在于不能持续以恒。如何才能持续以恒，即使外界的压力再大，对一个人的影响还是有限的。浩为的观点是：人必须认清自己，方能激发自己的潜力，再辅以一定的计划及追踪手段，如使用 FreeMind 软件或思维导图的思想来分析问题及实现“日程管理”，坚持一段时间，就能发现自己的思维习惯及日程等大有改善。

2.2 思维导图

有人说思维导图让你在一天之内分析几本书并牢记重点！随着思维导图使用的增多，思考问题的条理性大有提高，在无形中就提高了你的思考能力，更激发了你的思维潜能。详情：<http://www.google.cn/search?hl=zh-CN&num=100&q=思维导图+功能>

思维导图是一种革命性的思维工具。思维导图采用图形化的方式来展现想法和概念。作为一个可视化思考工具，它有助于信息结构化，便于更好的分析、理解、综合、回忆以及产生新的想法。正像所有的好点子一样，它的能力就在于它的简单，却又极其有效！

与传统的记笔记或者直线型文字不同，思维导图里的信息，完全可像大脑实际思考过程那样进行组织。思维导图兼具分析性和艺术性，会让大脑思维更为发散，可以激发所有认知功能。同时，最棒的地方是思维导图非常有趣。

19 世纪 60 年代英国著名心理学家东尼·博赞在研究大脑的力量和潜能过程中，发现艺术家达·芬奇在他的笔记中使用了许多图画、代号和连线。他意识到，那就是达芬奇拥有超级头脑的秘密所在。在此基础上，博赞发明了思维导图这一风靡世界的思维工具。

思维导图作用的概括：增强使用者的超强记忆能力、增强使用者的立体思维能力（思维的层次性与联想性）、增强使用者的总体规划能力。

为什么思维导图功效如此强大？道理其实很简单。

首先，它基于对人脑的模拟，它的整个画面正像一个人大脑的结构图（分布着许多“沟”与“回”）；

其次，这种模拟突出了思维内容的重心和层次；

第三，这种模拟强化了联想功能，正像大脑细胞之间无限丰富的连接；

第四，人脑对图像的加工记忆能力大约是文字的 1000 倍。

思维导图是一种创造性的和有效的记笔记的方法，能够用文字将你的想法“画出来”，让你更有效地把信息放进你的大脑，或是把信息从你的大脑中取出来。

所有的思维导图都有一些共同之处：它们都使用颜色，它们都有从中心发散出来的自然

结构，它们都使用线条、符号、词汇和图像，遵循一套简单、基本、自然、易被大脑接受的规则。使用思维导图，可以把一长串枯燥的信息变成彩色的、容易记忆的、有高度组织性的图画，它与我们大脑处理事物的自然方式相吻合。

那么思维导图是什么样子呢？请前面如图 28-1 所示的思维导图范例，采用可视化方式展示了人生需要引导的相关概念，再对发散出来的相关概念进行分析，很快就能分析出各点的内容及相关性。

尽管在刚开始的时候看起来太乱，不过一旦你放弃那根深蒂固的线性笔记的习惯，你就不会回头了。再看看有人用思维导图做的书本重点内容图：



(图 2-1 思维导图分析书本重点内容)

好处和用法

基本上来说，思维导图摆脱了呆板的线性思考，唤醒创造力，使做笔记又成为一种乐趣。然而我们能用思维导图做什么？

记笔记、头脑风暴（个人或者团队）、解决问题、学习和记忆、规划、研究和强化众多源头的信息、展现信息、洞察复杂对象、唤醒你的创造力，还有更多……

很难准确地说出来思维导图用法的数量，但思维导图的确能够有助于理清思考任何事情、不同情景的思路：个人、家庭、教育或者事业。计划每天生活或者规划一生、做读书笔记、启动项目、规划和撰写报告书，撰写博客。

也就是说，思维导图适用于任何事情。

思维导图画法

思维导图的画法就像 1-2-3 一样简单：

- * 首先在一个空白纸的中间地方把你想要发散的概念写出来或者画出来，建议将纸张横向使用。
- * 围绕这个中心主题发散相关副主题，并将每个副主题和中心用一条线连接起来。
- * 重复同样的过程不断逐级产生副主题，将这些与对应的副主题连接起来。

一些建议：

* 大量使用颜色，图形、符号。尽可能可视化出来，大脑会感谢你。很多不愿尝试的人，借口说他们不是“艺术家”，别让这个借口成为不去尝试的理由。

* 主题标签尽可能的简短，用一个字或者更好就用一个图片。特别是你第一次使用思维导图的时候，总是极其想用用一个完整的短语。不过不断找机会缩短成单词或者一个图形，那

样你的思维导图将会更为有效。

* 改变文字大小、颜色和排列，改变线条的粗细和长度等。尽你所能，多用可视化元素来强调重要的地方。每一个小地方都会激励大脑。

总结：思维导图绝对是一种令人着迷以及丰富的主题，这里也仅仅是说了点皮毛而已。如果你想更多的参考材料，Google 一把吧。

目前常用的软件为 FreeMind 和 MindManager，前者为开源软件，后者为商业软件。

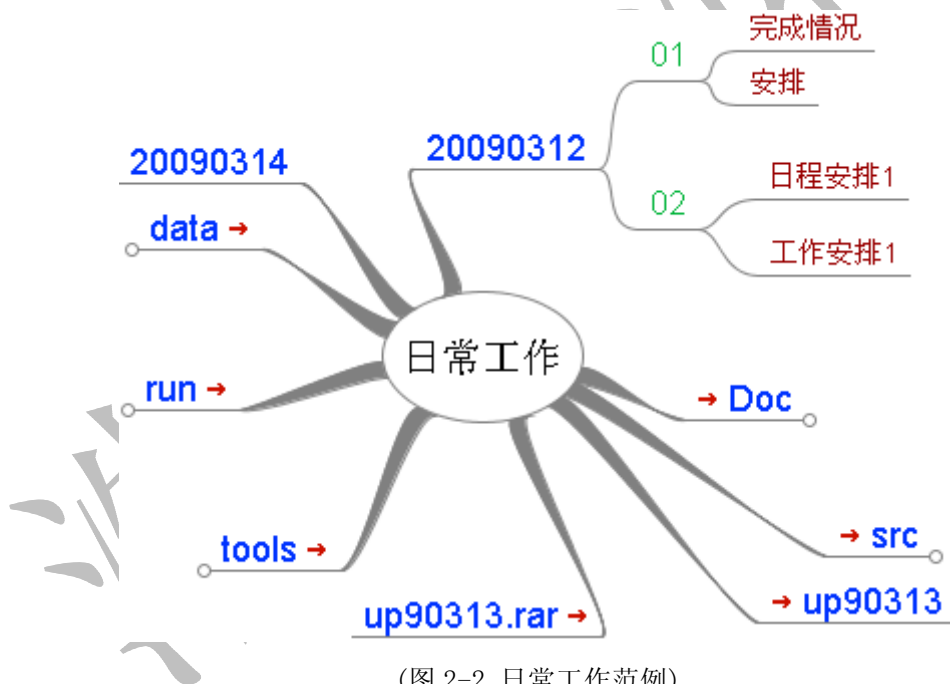
2.3 FreeMind 应用范例

一、日常工作目录

请浏览“1.3 Java 开发整合包使用说明”中 图 1-8 Work 目录结构 相关内容。

在日常工作中使用如此结构的目录，将有助于工作的顺利开展，形成一种良好的工作习惯。比如日常文件的管理，养成按日期存放及整理的习惯。以后要找文件时，不会找不着。

二、日程管理



(图 2-2 日常工作范例)

日程管理范例，见 20090312 节点，可以事前安排日程、工作，事后跟踪。已导入几个目录：20090314、Doc、data、src 等，可以直观便捷的查看目录及文件。

有个将在 2011 年考研究生的大三学生问我“考研复习，每天的各个时间片和对应要完成的任务可不可以做成思维导图？”我简单说了一下，他就明白了，试着画了一张，然后截图给我看。以下是整理后的对话：（他签名为“天行健，君子以自强不息！”，简称为自强）

自强：感觉就用到几个功能，先把大体轮廓做好了，我感觉应该往后细分，越详细越好。

浩为：感觉到思维导图的便利性了吧，往后细分后抓出重点。

自强：一个考研成功的师哥告诉我习惯的重要性，拿出点时间来制定计划，肯定比盲目

往下学有好处。

浩为：不仅是考研，以后的工作、生活更是如此。

自强：暑假的时候，就应该学会这个东西。（注：他曾参与浩为公司在暑假发起的“软件工程术语中英对照案例解析教程”项目）

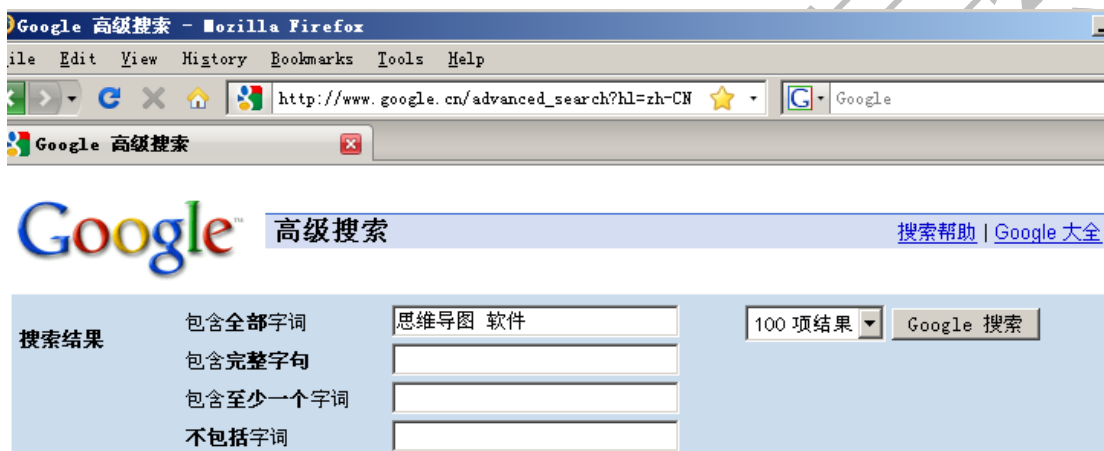
浩为：现在学会还来得及。

2.4 学会 Google 快速查找知识

当今社会，知识成爆炸式增长，即使你一刻不停地学习，也无济于事，怎么办？学以致用，当你需要什么，就去查找，怎么查？

查找知识，建议使用 google.cn 的高级搜索，地址如下：

http://www.google.cn/advanced_search?hl=zh-CN



(图 2-3 Google 快速查询)

选择 100 项结果，在 **包含全部字词** 处输入内容，这样可以快速查到自己的内容。如果显示的记录太多，可在输入的内容处加限制。

例：查“思维导图”，发现记录特多；则可限制为“软件”的记录，输入“思维导图 软件”，注意中间用空格隔开，则很快就能查到需要的内容。

为啥要选 100 项，因为选 100 项查找更方便。如果你用默认的 10 项，当你看完 10 项结果，如没有你想要找的内容，你得看第 2 页、第 3 页，甚至上 10 页，如果选择显示 100 项呢，一页就有 100 项结果，再说还可以在 100 项结果里对比，到底哪项是你要找的内容。

依我的经历，一般大多在第 1 页就能查到我要的东西，当然也有不少要到第 2 页，甚至更多页才能找到。

建议将某次选择 100 项后的查询作为一个书签，这样下次打开时就不用再选了。

我现在的书签如下：

<http://www.google.cn/search?hl=zh-CN&num=100&q=howwe>

直接出来 100 项，修改成待查询的词即可。

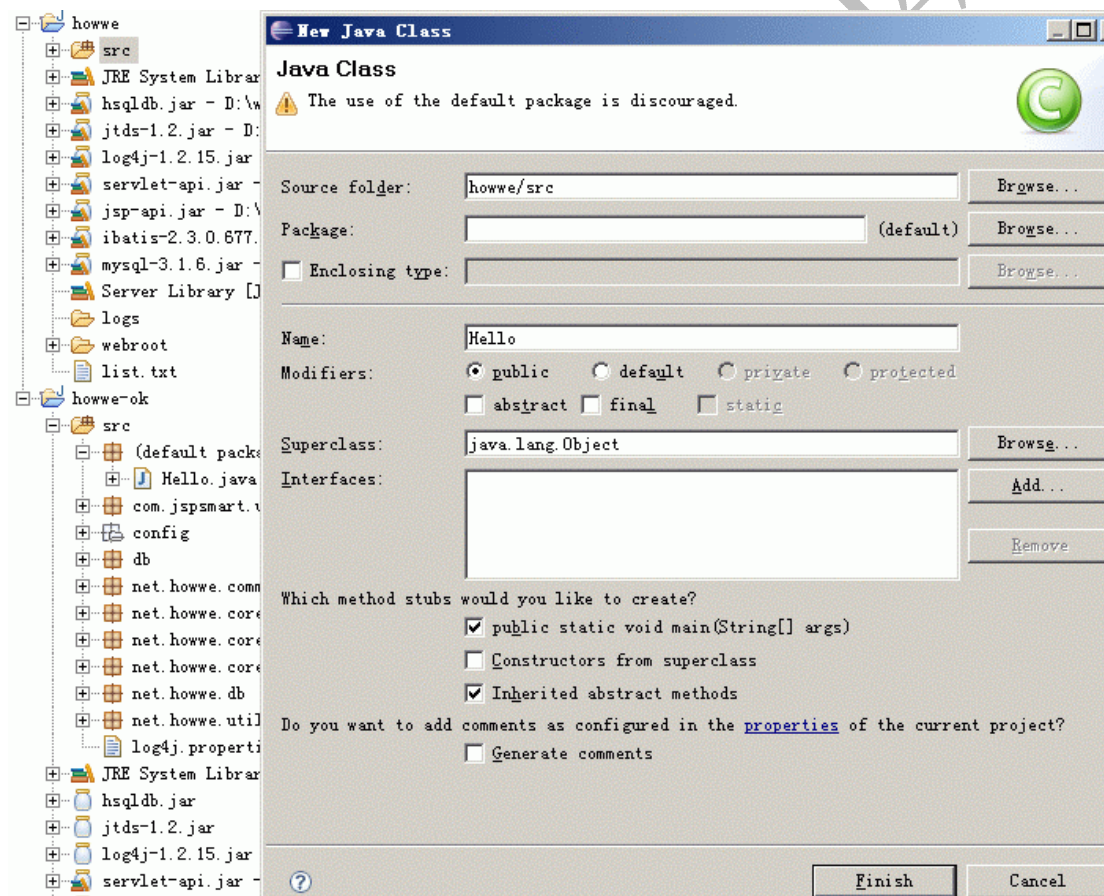
第 3 章 Hello World 范例

我们先从最简单的代码入手，输入一个字符串“Hello World”，而字符串就是一种数据类型。再输出多个。为啥能输出多个，是因为我们定义了变量及使用了循环结构。其中循环结构就是控制结构的一种。

变量为啥能定义，也是因为程序语言定义了数据类型。绝大部分程序都用来处理数据，处理数据时，处理的对象就是已结构化的多种数据类型的组合体，这个组合体一般称为数据结构。所以程序语言的三大基本知识点分别是数据类型、控制结构、数据结构。

3.1 输出单一字符串

先新建一个最简单的程序，我们将创建 Hello 类。

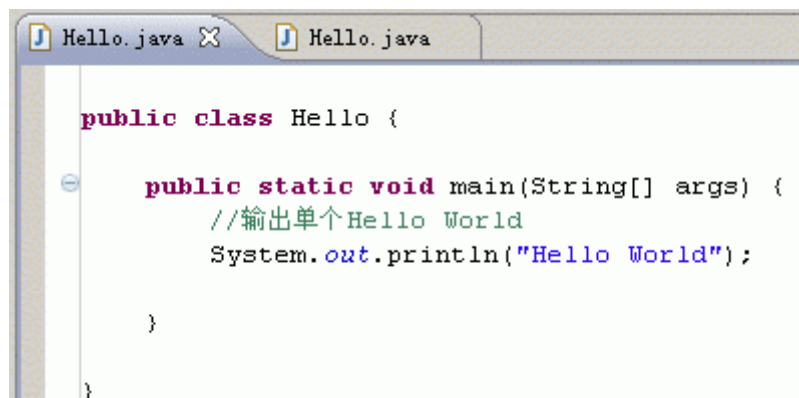


(图 3-1 创建 Hello 类)

如图 3-1 所示，在 howwe 项目 src 目录上单击右键，然后选择 **New => Class**（中文版对应为 **新建 => 类**）。在 Name 处键入 Hello 作为类名称。在“Which method stubs would you like to create?”下面，选中“public static void main(String[] args)”复选框，然后按 **Finish**。

这样将在编辑器区域创建一个包含 Hello 类和空的 main() 方法的 java 文件。在 main() 方法中添加如下代码，保存后如如图 3-2 所示。

```
//输出单个Hello World
System.out.println("Hello World");
```



(图 3-2 编辑器中的 Hello 类)

在键入时能注意到 Eclipse 编辑器的一项特性：代码自动完成。当键入括号或双引号时，Eclipse 会自动提供配对的符号，并将光标置于符号对之内。

在其他情况下，可以通过按 Ctrl-Space 来调用代码自动完成功能。代码自动完成提供了上下文敏感的建议列表，你可通过键盘或鼠标来从列表中选择。这些建议可以是针对某个特定对象的方法列表，也可以是基于不同的关键字（比如 for 或 while）来展开的代码片段。

说明：System.out.println 用于在打印内容，然后终止该行。

具体可以查看 Java API 开发人员文档中 System 的说明。

Java API 下载地址：<http://developers.sun.com.cn/Java/list.html>

更多内容也可用 google.cn 的高级搜索，地址如下：

http://www.google.cn/advanced_search?hl=zh-CN

代码：System.out.println("Hello World"); 中的 "Hello World" 是一个字符串常量，什么是字符串，字符串是一种数据类型。什么是常量，值不变的量称为常量。习惯上将常量的名字统统大写。

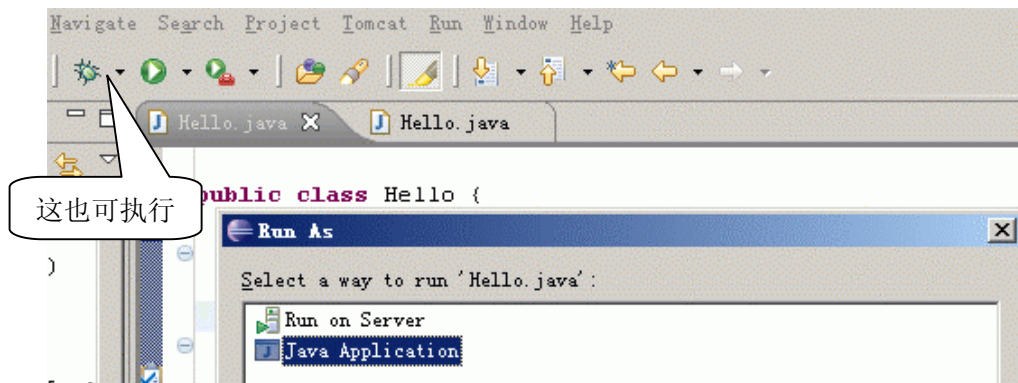
3.2 执行代码

一旦代码无错误地编译完成，你就能够从 Eclipse 菜单上选择 Run 来执行该程序。

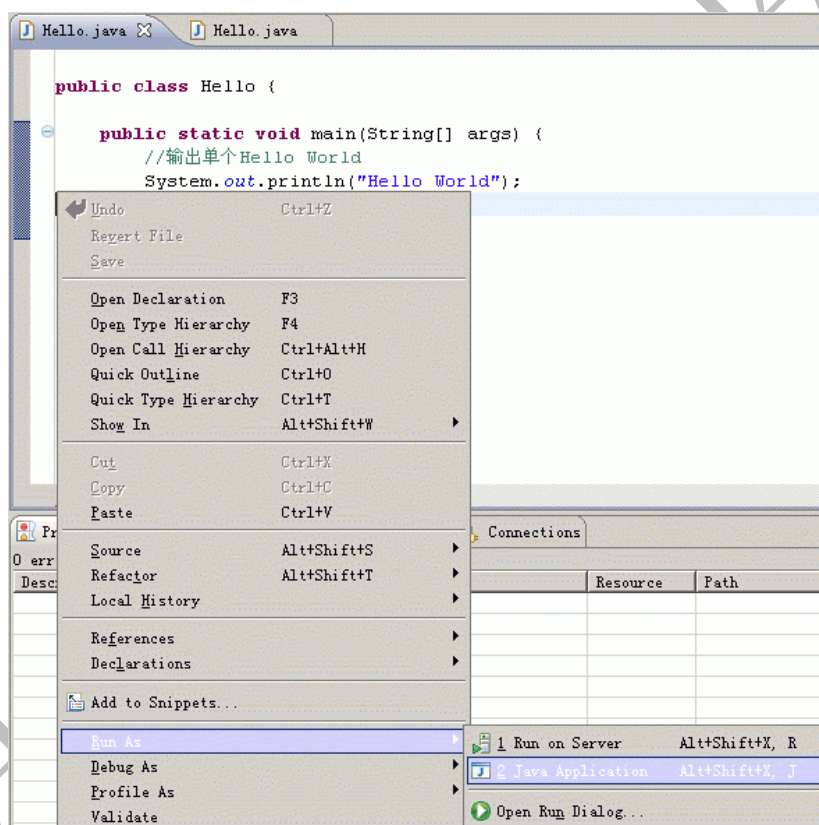
注意：这里不存在单独的编译步骤，因为编译是在保存代码时进行的。如果代码没有语法错误，它就可以运行了。

有三种执行方式可供选择：

- 从菜单 Run 选择 Run As
 - 点击如图 3-3 所示的按钮，选择 Run As
 - 在文件编辑区，单击右键，在如图 3-4 所示，选择 Run As
- 再选择 Java Application

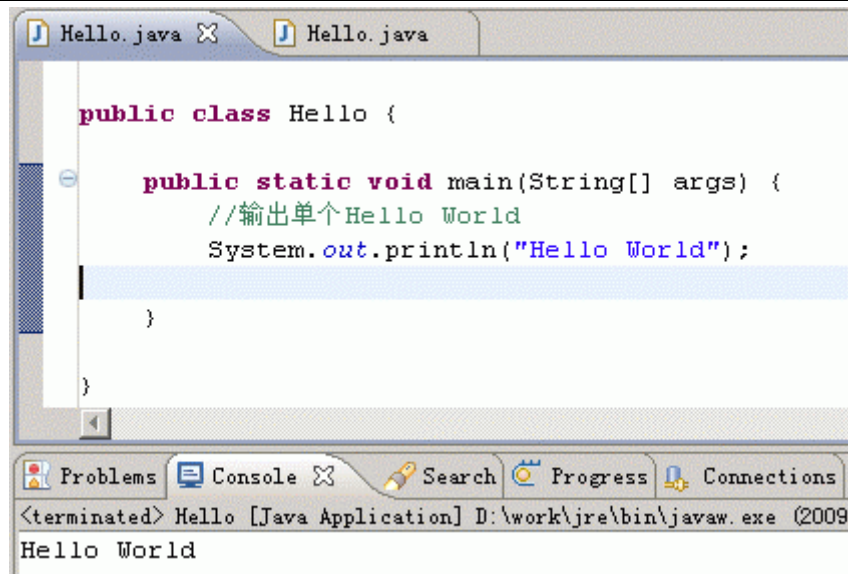


(图 3-3 执行 Hello)



(图 3-4 右键执行 Hello)

运行后，在控制台区将出现一个标签 Console，显示了程序的输出内容，如图 3-5 所示。



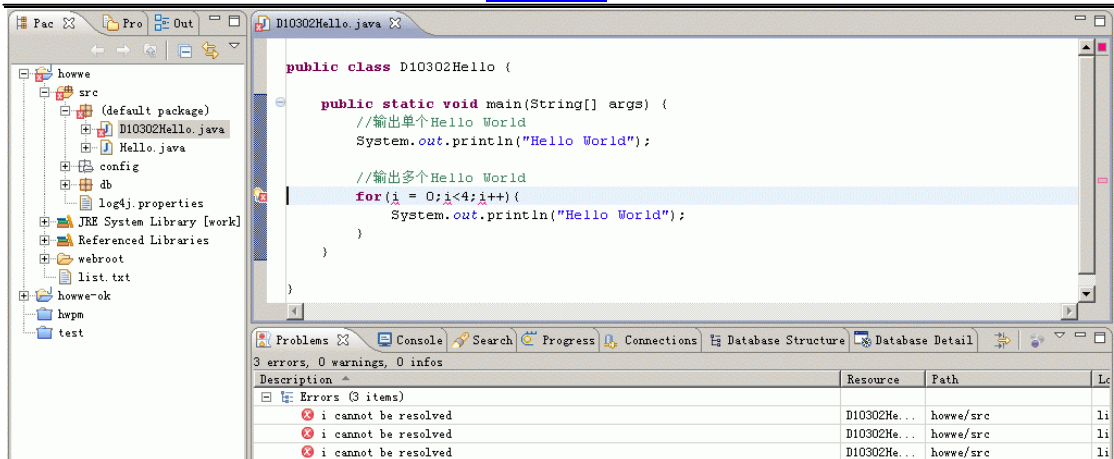
(图 3-5 执行 Hello 后的输出)

3.3 输出多个字符串

在 howwe 项目 src 目录上单击右键，然后选择 **New=>Class**（中文版对应为 **新建=>类**）。在 Name 处键入 D10302Hello 作为类名称。在 “Which method stubs would you like to create?” 下面，选中 “public static void main(String[] args)” 复选框，然后按 **Finish**。

这样将在编辑器区域创建一个包含 D10302Hello 类和空的 main() 方法的 java 文件。在 main() 方法中添加如下代码，保存后如如图 3-6 所示（注意其中 i 的声明是有意省略了的）。

```
//输出单个Hello World  
System.out.println("Hello World");  
  
//输出多个Hello World  
for(i = 0;i<4;i++){  
    System.out.println("Hello World");  
}
```

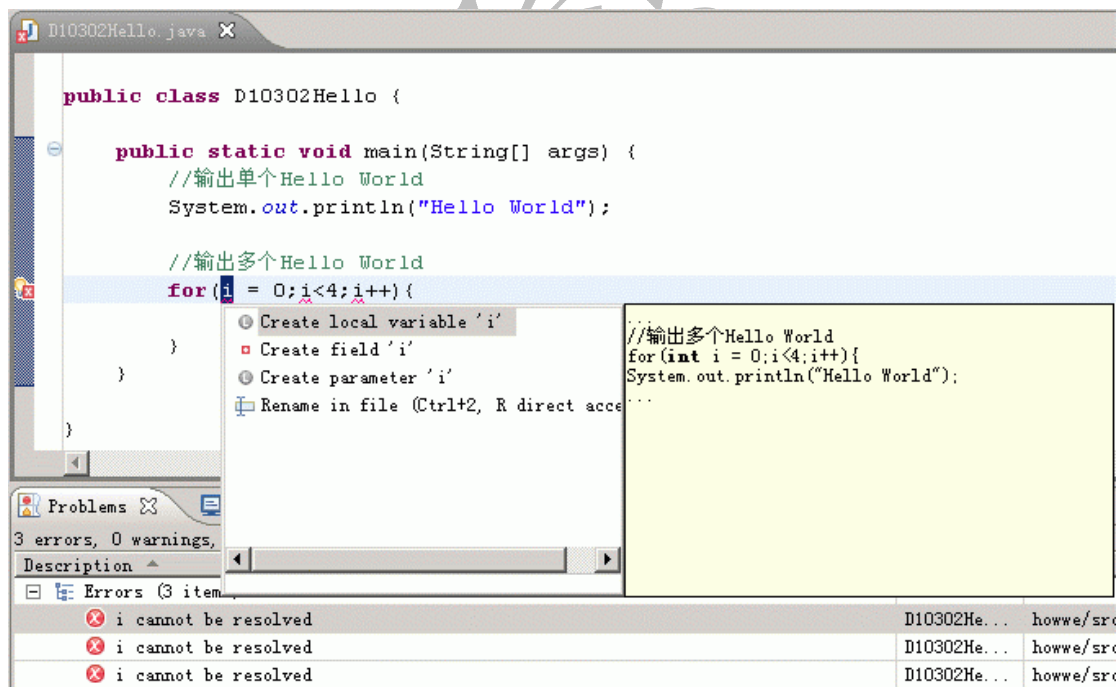


(图 3-6 编辑器中的 D10302Hello 类)

保存代码后，能看到 Eclipse 编辑器的一项特性：语法检查。

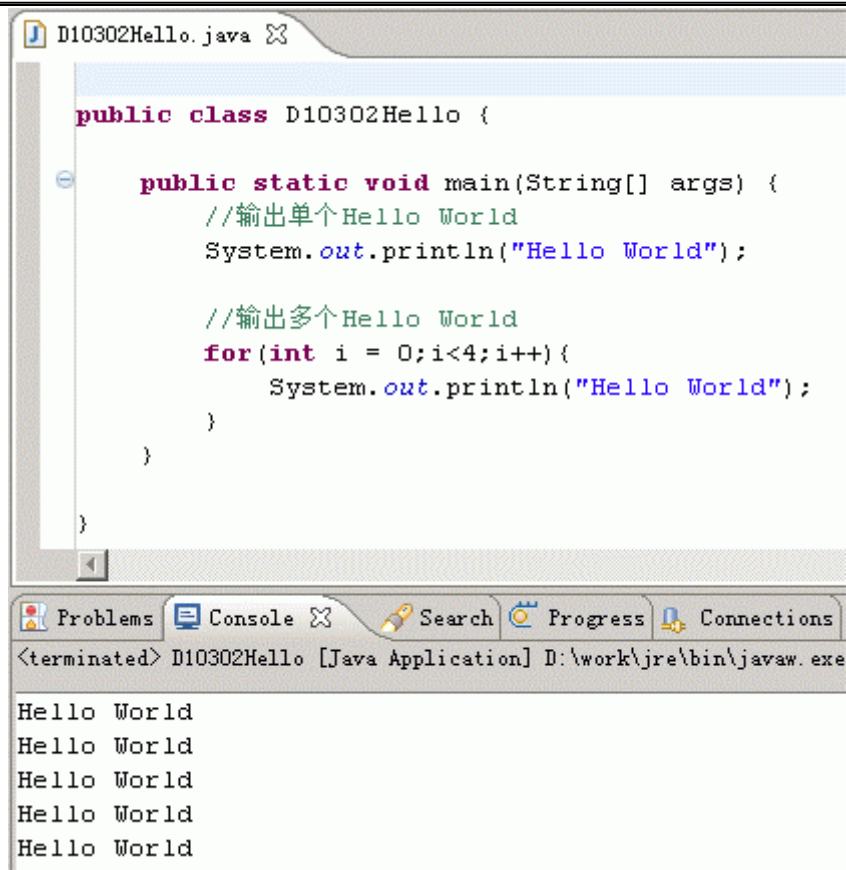
语法检查依赖增量编译。每当保存代码，它就在后台接受编译和语法检查。默认情况下，语法错误将以红色下划线显示，一个带白“X”的红点将出现在左边沿。其他错误在编辑器的左边沿通过灯泡状的图标来指示；这些就是编辑器或许能为你修复的问题——即所谓的 Quick Fix（快速修复）特性。

上面的代码例子在 for 语句后面有一个灯泡状图标，如图 3-6 所示，因为 i 的声明被省略了。双击该图标将调出建议的修复列表。在此例中，它将提供创建一个类字段 i、一个局部变量 i 或一个方法参数 i 的建议；单击其中的每一个建议都会显示将要生成的代码。图 3-7 显示了该建议列表和建议创建一个局部变量之后生成的代码。



(图 3-7 Quick Fix 建议)

双击该建议就会把建议代码插入到代码中的恰当位置，保存后按 3.2 所叙的执行方法程序，执行结果如图 3-8 所示，一共输出 5 个“Hello World”字符串。



(图 3-8 执行 D10202Hello)

代码 `int i = 0;` 其实就是定义了一个 `Int` 变量。什么是变量，又与常量有什么区别，以下是两者的简单对比，详情请参考“第 4 章 数据类型”。

常量与变量的区别：

Java 常量名用字符或字符串表示，它区分类型，如整型常量 123，实型常量 1.23，字符常量 'a'，布尔常量 true、false 以及字符串常量 "This is a constant string."。

与 C、C++ 不同，Java 中不能通过 `#define` 命令把一个标识符定义为常量，而是用关键字 `final` 来实现，如 `final double PI = 3.14159`。

变量是 Java 程序中的基本存储单元，它的定义包括变量名、变量类型和作用域几个部分。

在 `D10302Hello` 类中，先执行语句：

```
//输出单个Hello World  
System.out.println("Hello World");
```

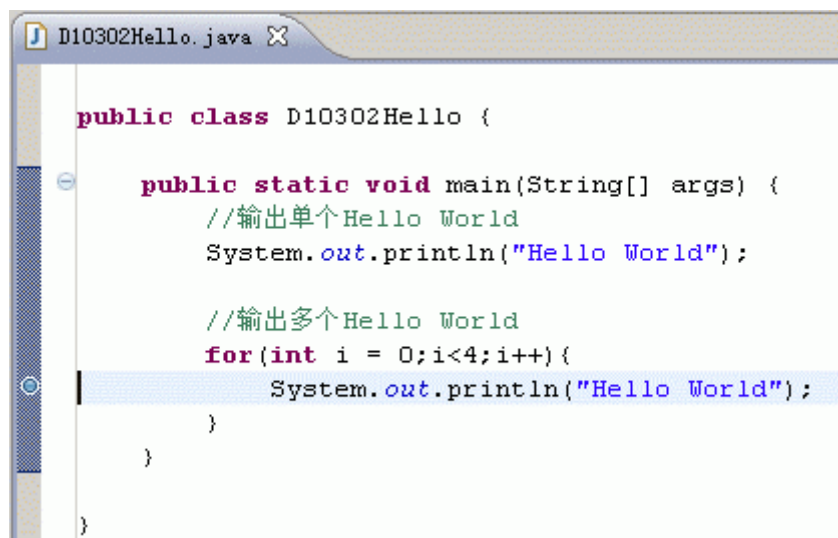
再执行代码段：

```
//输出多个Hello World  
for(i = 0;i<4;i++){  
    System.out.println("Hello World");  
}
```

这两者按顺序执行，从控制结构上来说是一种顺序结构。但后面的代码段循环执行了四次，属于循环结构。控制结构的详细内容请浏览“第 5 章 控制结构”。

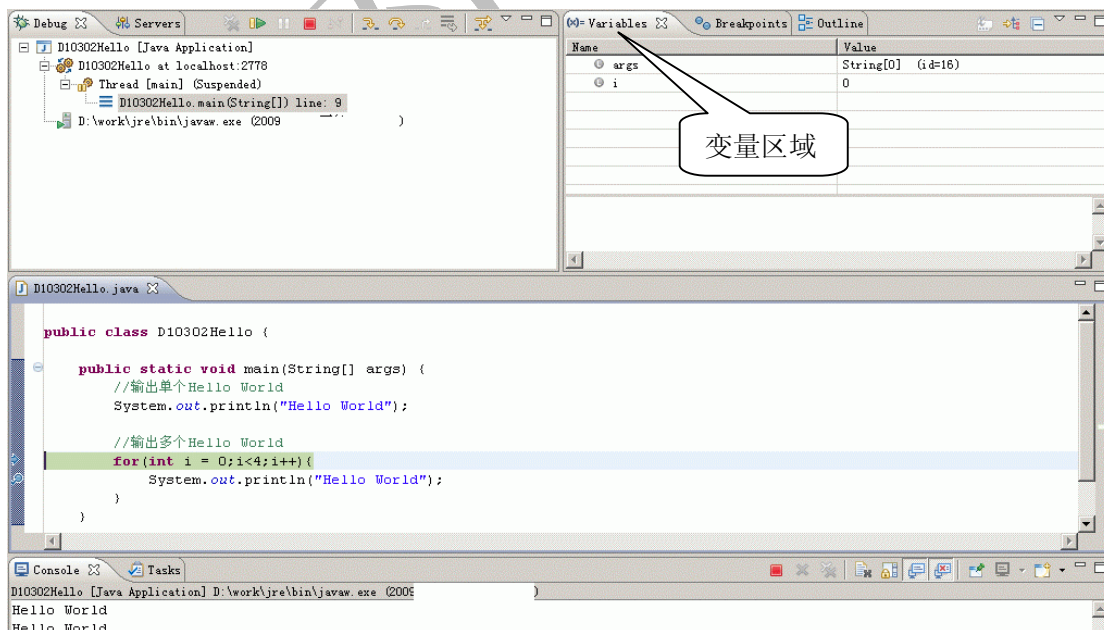
3.4 调试代码

可以在 Java 调试器中运行程序。在 main() 方法中的第二个 System.out.println 处，双击编辑器视图左端的灰色边沿，即在此处设置一个断点。一个蓝色的点将会出现在那里，图 3-9 所示。



(图 3-9 设置断点)

然后从 Run 菜单上选择 Debug。参照 3.2 节的描述，只是将 Debug 代替 Run，在选择 Java Application 之后，透视图将自动切换到 Debug 透视图，其中具有许多有趣的新视图，如图 3-10 所示：



(图 3-10 Debug 界面)

首先，请注意该透视图左上角的 Debug 视图。这个视图显示调用堆栈，并且标题栏中

有一个工具栏，它允许您控制程序的执行，包括继续、挂起或终止程序、跟踪下一个语句、单步执行下一个语句，或者从方法返回。

快捷键说明：

F6 单步执行语言；

F8 执行到下一次断点或结束（没下一断点就结束），也称为 Resume。

右上角的窗格包含许多选项卡式的视图，包括 Variables、Breakpoints、Expressions 和 Display。这里我单击了 Variables 视图，以便我们能够看到 i 的当前值。

可以通过上下文敏感的帮助，获得关于这些视图的更多信息：单击视图的标题，然后按 F1。

调试结束，如果在 Console 出现如下内容：

```
ERROR: JDWP Unable to get JNI 1.2 environment, jvm->GetEnv() return code = -2
JDWP exit error AGENT_ERROR_NO_JNI_ENV(183): [../../src/share/back/util.c:820]
```

请在代码最后加入以下语句：System.exit(0)即可，注意，只需要在这种情况下加，在正式运行时不需要。

第 4 章 数据类型

IT 知识点数量很多，即使你一刻不停的学，一辈子也学不完。那该怎么办呢？万变不离其宗，每门学科都有自己的基本原理。了解基本原理，其他东西就可以举一反三，从而达到事半功倍的效果。

讲述 SSH 的书籍很多，教别人怎么使用的人更多，可有几个人能真正讲明白。这些人除了会教别人通过工具这个怎么配、那个怎么配之外，有几个能讲述 SSH 的基本原理。简单来说，SSH 以 Struts 为核心，而 Struts 呢，只是一个访问控制器加一堆的标签及 FormBean。

我们开发项目，整体思想是第一位，开发语言的基本知识是第二位。该掌握哪些基础知识呢？数据类型、数据结构、控制结构是基础的基础。

数据类型是编程的第一步，到底什么是数据类型？数据类型是用来约束数据的解释。数据类型描述了数值的表示法、解释和结构，以及算法操作，或是物件在内存中的储存区，或者其它储存装置。

程序的目的简单来说就是运算，除了运算还是运算。有以下几种运算：算术、逻辑、比较、递增、递减。

4.1 计算机数据类型

所有计算机中的数据，最基本的单位是比特 (bit)，也叫位，即一个二进制数据，用 0 或 1 表示。资料的最小寻址单位，称为字节（通常是 8 bit，以八个位为一组）。在同一时间中计算机能处理二进制数的位数叫字长。32 位计算机在同一时间内处理字长为 32 位的二

进制数据。64 位计算技术从 2004 年推出至今，目前还没有大规模应用。

字节是计算机信息技术用于计量存储容量和传输容量的一种计量单位，1 个字节等于 8 位二进制。字节，英文名称是 Byte，Binary Term 的缩写。它也是程序设计语言里不可缺少的基本数据类型——整数（Int）。Byte（字节）可被缩写成 B，例如 MB 表示 Megabyte。

此类单位的换算为：

1 bit = 1 位/比特（二进制数据）
1 byte = 8 bit（字节）（byte，Binary Term）
1 字母 = 1 byte = 8 bit
1 汉字 = 2 byte = 16 bit
1 Kb = 1024 byte（2 的 10 次方字节）（Kb，Kilobyte）
1 Mb = 1024 Kb（2 的 20 次方字节）（Mb，Megabyte）
1 Gb = 1024 Mb（2 的 30 次方字节）（Gb，Gibabyte）
1 Tb = 1024 Gb（2 的 40 次方字节）（Tb，Terabyte）



(图 4-1 计量单位的换算)

4.2 常见数据类型

在编程语言中，常见的数据类型为原始类型（如：整数、浮点数或字符），对 Java 来说，叫简单数据类型。Java 不支持 C、C++ 中的指针类型、结构体类型和共用体类型。

每一个数据类型都有一个数值上的最大和最小值，称作数值范围。了解数值范围很重要，当你使用较小的型别时，只能储存范围之内的数值。试图储存一个超出其范围的数值，可能会导致编译或执行错误，或者不正确的计算结果（因为被截断）。

程序员可以利用多种数据类型：某些由编程语言定义，某些由外部库定义，还有些则由程序员来定义。编程语言提供若干原始数据型别，作为程式以及专用化复合型别的建立基础。很多编程语言都依赖于特定的计算机类型和对数据类型属性的具体编译实现，但 Java 虚拟机已经定义了内置数据类型。这意味着不管 Java 虚拟机（JVM）运行在何等级别的操作系统之上，数据类型的属性都是完全一样的。

1、简单数据类型

简单数据类型是不能再简化的、内置的数据类型，由编程语言定义，表示真实的数字、字符和整数。更大、更复杂的数据类型可以采用简单数据类型的组合来定义。在大多数情况下，简单数据类型都具有其硬件等价物。比方说，int 简单类型有时存放在 32 位硬件寄存器内。Java 提供了几类简单数据类型表示数字和字符。

简单数据类型通常划分为以下几种类别：实数、整数、字符和布尔值。实数类别分两种：

float、double，整数类别分四种：byte、short、int 和 long，字符类别一种：char，布尔值类别一种：boolean。下表详细列出了 Java 的简单数据类型。

简单类型	大小	范围/精度	默认值
int	4 字节	-2,147,483,648 (-2^{31}) 到 2,147,483,647 ($2^{31}-1$)	0
byte	1 字节	-128 到 127 (2^7-1)	0
short	2 字节	-32,768 到 32,767 ($2^{15}-1$)	0
long	8 字节	-9,223,372,036,854,775,808 (-2^{63}) 到 $2^{63}-1$	0
char	2 字节	整个 Unicode 字符集，16 位无符号型数据	'u0000'
boolean	1 位	true 或者 false	false
float	4 字节	32 位 IEEE 754 单精度	0.0
double	8 字节	64 位 IEEE 754 双精度	0.0

Java 中的所有数字变量都是有符号的，Java 不允许数据类型之间随意转换。只有数字变量之间可以进行类型转换。如 boolean 不能转换为其他数据类型，而且其他数据类型也不能转换为 boolean。因为 Java 的简单数据类型都经过准确定义，并且不允许直接访问内存，所以 Java 没有 sizeof 运算符。

Java 的简单数据类型不是对象，提供了 Integer、Byte、Short、Long、Boolean、Character、Float、Double 等内置的封装类。这些封装 (wrapper) 类提供了很直观的实用方法。具体可以查看 Java API 开发人员文档中 java.lang 的说明。

下载地址：<http://developers.sun.com.cn/Java/list.html>

(题外话：Sun 的服务意识不是一般的差，竟连这个网址 http://www.java.com/zh_CN 上面都没有 API 的下载地址)

对象、类等概念请参考《从人生需要引导论 Java 快速开发》“第 22 章 面向对象”。

在 Java 语言中，简单数据类型作为类的成员变量，声明时自动初始化为默认值，除非显式地声明。简单数据类型为某一方法声明局部变量时不会自动地初始化，并且会导致编译器扔出类似以下的错误消息 “Variable x may not have been initialized. (x 变量没有初始化)”，请注意初始化。

2、索引引用复合数据类型

(注：初学者可略读，待需要时再细读。因为部分内容已涉及面向对象的思想：如类、接口，但数组必看。)

同简单数据类型的定义一样，JVM 定义了索引 (reference) 这种数据类型。索引类型可以“引用”变量，由于 Java 没有明确地定义指针类型，所以索引类型可以被认为就是指向实际值或者指向变量所代表的实际值的指针。一个对象可以被多于一个以上的索引所“指”。JVM 从不直接对对象寻址而是操作对象的索引。

索引类型分成三种：类 (class)、接口 (interface) 和数组 (array)。索引类型可以引用动态创建的类实例、普通实例和数组。故结构 (struct) 或联合 (union) 数据类型可以通过类或者接口来构造，类提供了捆绑数据和方法的方式，同时可以限制对类的私有数据的访问。索引还可以包含特殊的值，null 索引。null 索引在运行时上并没有对应的类型，但它可以被转换为任何类型。索引类型的默认值就是 null。

3、类 (Class)

类指的是定义方法和数据的数据类型。从内部来看，JVM 通常把 class 类型对象实现为指向方法和数据的一套指针。定义 class 类型的变量只能引用类的实例或者 null，如以下代码所示：

```
MyObject anObject = new MyObject(); // 合法
MyObject anotherObject = null; // 合法
MyObject stillAnotherObject = 0; // 非法
```

4、接口 (interface)

接口好比一种模版，这种模版定义了对对象必须实现的方法，其目的就是让这些方法可以作为接口实例被引用。

接口不能被实例化。类可以实现多个接口，并且通过这些实现的接口而被索引。接口变量只能索引实现该接口的类的实例。比方说，假设我们定义了一个接口，名字是 Comparable，同时还定义了一个类 SortItem，这个类实现了接口 Comparable，那么我们可以编写以下的定义代码：

```
Comparable c = new SortItem();
```

假如 Comparable 接口定义了一个方法：public void compare(Comparable item)，那么 SortItem 类就必须提供 compare 方法的实现，如以下代码所示：

```
public class SortItem implements Comparable
{
    public void compare(Comparable item)
    {
        ...method implementation here
    }
}
```

高级用法：相关概念的对比

interface：是一组规范，你要 implements 就必须实现它的所有方法，可以多重实现，增加了语言的灵活性。

class：是一种层次关系，可以重写和继承，多态就是通过这种方式实现的。

extends：继承、扩展的意思，实现父类的特性。在 Java 中继承只能是一次，不可以实现多重继承，这和 c/c++或其他一些语言有区别。即 extends 只能实现单继承（也就是一个儿子只能有一父亲）。

implements：实现接口的意思。功能类似于 extends，都是为了扩展自身的一些功能，但不同的是，implements 实现接口可以是多个（implements 后面可以接多个接口），implements 可以是多重继承（好比一个人可以有多个老师）。从实际使用来讲，implements 功能要更强大一些，但两者都很重要，值得深究。

extends 是继承，每个类只能从一个父类继承；implements 是实现接口，每个类可以实现多个接口。对于 class 而言，extends 用于（单）继承一个类（class），而 implements 用于实现一个接口（interface）。interface 的引入是为了部分地提供多继承的功能。

在 interface 中只需声明方法头，而将方法体留给实现的 class 来做。

那些实现 interface 的 class 实例完全可以当作 interface 的实例来对待。

5、Java 数组 (array)

Java 数组是动态创建的索引对象，这一点和类非常相似，此外，同类一样，数组只能索引数组的实例或者 null，如以下代码所示：

```
int[] myIntArray = new int[5];
int[] anotherIntArray = null;
```

数组是 Object 类的继承，这样，Object 类的所有方法都可以被数组调用。数组对象由元素组成。元素的数目也可以为 0，在这种情况下称做数组为空。

所有的数组都是从 0 开始对元素编号的，这意味着数组内的第 1 个元素的索引编号是数字 0。所有对数组元素的访问都会在运行时上接受检查，如果试图使用编号小于 0 或者大于数组长度来索引元素就会产生 Array Index out of Bounds Exception 异常并被抛出。

数组的元素按整型值索引，如以下代码所示：

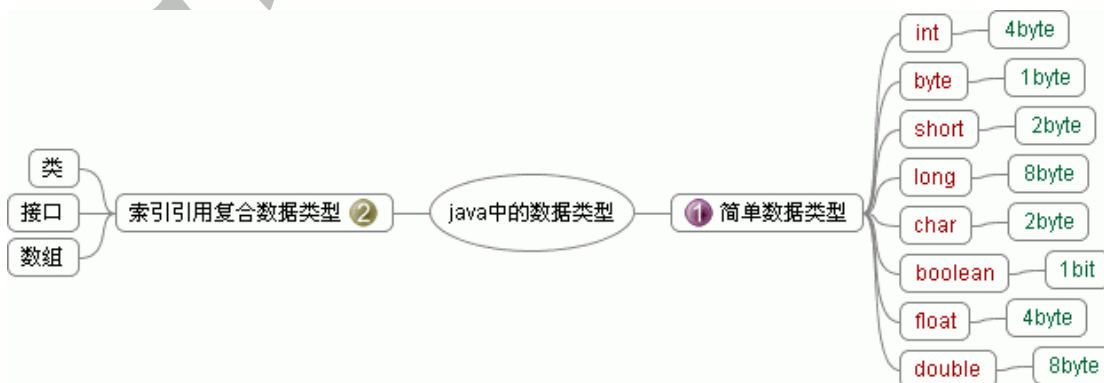
```
int[] myIntArray = { 9, 5, 6 };
int int1 = myIntArray[0];
int int2 = myIntArray[1];
int int3 = myIntArray[2];
```

数组对象的长度是不变的。为了改变数组变量的长度，你必须创建另一个数组并赋给变量，如以下代码所示：

```
int[] myIntArray = { 9, 5, 6 };
System.out.println("myIntArray length = " + myIntArray.length); // 输出结果是 3
myIntArray = new int[] { 3, 6, 4, 2, 8 };
System.out.println("myIntArray length = " + myIntArray.length); // 输出结果是 5
```

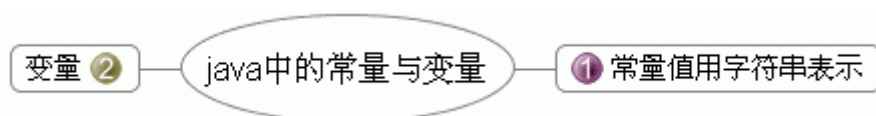
复合数据类型由构造器内定义的代码实现初始化，构造器在类的创建函数（比如“new”之类）应用到类变量的时候被调用。构造器被调用之前，类的各个成员被初始化为各自的默认值或者显式赋予的初始值。

Java 语言包含了大量的预定义复合数据类型。其中之一的 String 类属于 java.lang 包。String 类提供的方法执行常用的字符串操作，比如 length()、substring(int beginIndex)、toUpperCase()、toLowerCase()、equals() 等等。另一种常用的 Java 复合数据类型是 Vector 类，它属于 java.util 包。Vector 类定义的方法执行可扩展对象数组的常用操作。比如 add(int index, Object element)、elementAt(int index)、isEmpty() 和 remove(int index) 等。这些就是一小部分预定义复合数据类型的代表。



(图 4-2 Java 数据类型)

4.3 定义常量与变量



(图 4-3 变量与常量)

一、常量

Java 常量值用字符串表示，它区分类型，如整型常量 123，实型常量 1.23，字符常量 'a'，布尔常量 true、false 以及字符串常量 "This is a constant string."。

与 C、C++不同，Java 中不能通过 #define 命令把一个标识符定义为常量，而是用关键字 final 来实现，如 final double PI = 3.14159。

二、变量

变量是 Java 程序中的基本存储单元，它的定义包括变量名、变量类型和作用域几个部分。

①变量名是一个合法的标识符，它是字母、数字、下划线或美元符 "\$" 的组合。

Java 对变量名区分大小写，变量名不能以数字开头，而且不能为保留字。变量名应具有一定的含义，以增加程序的可读性。

合法的变量名如: myName、value-1、dollar \$ 等。非法的变量名如: 2mail、room#、class(保留字)等。

②变量类型可以为上一节所说的任意一种数据类型。

③变量的作用域指明可访问该变量的一段代码。声明一个变量的同时也就指明了变量的作用域。下表为作用域的区别：注意不写时默认为 friendly

作用域	当前类	同一 package	子孙类	其他 package
public	√	√	√	√
protected	√	√	√	
friendly	√	√	×	×
private	√	×	×	×

按作用域来分，变量可以有下面几种：局部变量、类变量、方法参数、例外处理参数。

- 局部变量在方法或方法的一块代码中声明，它的作用域为它所在的代码块(整个方法或方法中的某块代码)。
- 类变量在类中声明，而不是在类的某个方法中声明，它的作用域是整个类。
- 方法参数传递给方法，它的作用域就是这个方法。
- 例外处理参数传递给例外处理代码，它的作用域就是例外处理部分。

在一个确定的域中，变量名应该是唯一的。通常，一个域用大括号 {} 来划定。

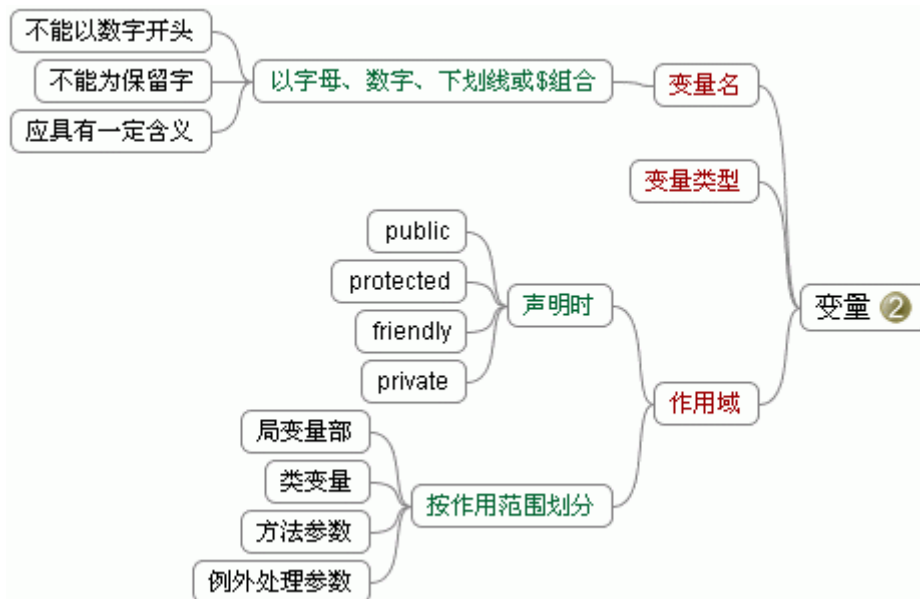
④变量的声明格式为：

```
type identifier[=value][, identifier[=value]... ];
```

例如：int a, b, c;

```
double d1, d2 = 0.0;
```

其中，多个变量间用逗号隔开，`d2 = 0.0` 对实型变量 `d2` 赋初始值 `0.0`，只有局部变量和类变量是可以这样赋值，而方法参数和例外处理参数的变量值是由调用者给出的。



(图 4-4 变量)

三、整型数据

1、整型常量：

与 C、C++ 相同，Java 的整常数有三种形式：

- ① 十进制整数，如 123，-456，0
- ② 八进制整数，以 0 开头，如 0123 表示十进制数 83，-011 表示十进制数 -9。
- ③ 十六进制整数，以 0x 或 0X 开头，如 0x123 表示十进制数 291，-0X12 表示十进制数 -18。

整型常量在机器中占 32 位，具有 `int` 型的值，对于 `long` 型值，则要在数字后加 `L` 或 `l`，如 `123L` 表示一个长整数，它在机器中占 64 位。

2、整型变量：

整型变量的类型有 `byte`、`short`、`int`、`long` 四种。

`int` 类型是最常使用的一种整数类型。它所表示的数据范围足够大，而且适合于 32 位、64 位处理器。但对于大型计算，常会遇到很大的整数，超出 `int` 类型所表示的范围，这时要使用 `long` 类型。

由于不同的机器对于多字节数据的存储方式不同，可能是从低字节向高字节存储，也可能是从高字节向低字节存储，这样，在分析网络协议或文件格式时，为了解决不同机器上的字节存储顺序问题，用 `byte` 类型来表示数据是合适的。但在通常情况下，由于 `byte` 表示的数据范围很小，容易造成溢出，应避免使用。

`short` 类型则很少使用，它限制数据的存储为先高字节，后低字节，这样在某些机器中会出错。

3、整型变量的定义，如：

```
byte b; //指定变量 b 为 byte 型
```

```
short s; //指定变量 s 为 short 型  
int i; //指定变量 i 为 int 型  
long l; //指定变量 l 为 long 型
```

四、浮点型(实型)数据

1、实型常量

与 C、C++相同，Java 的实常数有两种表示形式：

- ①十进制数形式，由数字和小数点组成，且必须有小数点，如 0.123, .123, 123., 123.0。
- ②科学计数法形式。如:123e3 或 123E3，其中 e 或 E 之前必须有数字，且 e 或 E 后面的指数必须为整数。

实常数在机器中占 64 位，具有 double 型的值。对于 float 型的值，则要在数字后加 f 或 F，如 12.3F，它在机器中占 32 位，且表示精度较低。

2、实型变量

实型变量的类型有 float 和 double 两种，双精度类型 double 比单精度类型 float 具有更高的精度和更大的表示范围，常常使用。

3、实型变量定义，如

```
float f; //指定变量 f 为 float 型  
double d; //指定变量 d 为 double 型
```

[注]与 C、C++不同，Java 中没有无符号型整数，而且明确规定了整型和浮点型数据所占的内存字节数，这样就保证了安全性和平台无关性。

五、字符型数据

1、字符常量

字符常量是用单引号括起来的一个字符，如 'a'、'A'。另外，与 C、C++相同，Java 也提供转义字符，以反斜杠(\)开头，将其后的字符转变为另外的含义。

与 C、C++不同，Java 中的字符型数据是 16 位无符号型数据，它表示 Unicode 集，而不仅仅是 ASCII 集，例如 u0061 表示 ISO 拉丁码的 'a'。

转义字符描述

```
\ddd 1 到 3 位 8 进制数据所表示的字符(ddd)  
\uxxxx 1 到 4 位 16 进制数所表示的字符(yyyy)  
\' 单引号字符  
\ 反斜杠字符  
\r 回车  
\n 换行  
\t 横向跳格
```

范例：

请参照 3.1 节中提到的建立类的步骤，在项目 Howwe 的 src 目录上建立类 D10401transfer，并把代码输入。

注意：其他章节中的范例，如没有特别说明，请参照 3.1 节中提到的方法建立类，不再讲述具体过程，即只给出范例的代码，请自己建立类并执行。


```
public class D10401transfer {  
    public static void main(String[] args) {  
        String s="0.1.2.3.4.5.6.7";  
        String[] str=s.split("\\.");  
        for(int i=0;i<str.length;i++)  
        {  
            System.out.println(str[i]);  
        }  
    }  
}
```

执行结果略。

2、字符型变量

字符型变量的类型为 char，它在机器中占 16 位，其范围为 0~65535。字符型变量的定义如：

```
char c = 'a'; //指定变量 c 为 char 型，且赋初始值为'a'
```

与 C、C++不同，Java 中的字符型数据不能用作整数，因为 Java 不提供无符号整数类型。但是同样可以把它当作整数数据来操作。

例如：

```
int three = 3;  
char one = '1';  
char four = (char)(three+one); //four='4'
```

上例中，在计算加法时，字符型变量 one 被转化为整数，进行相加，最后把结果又转化为字符型。

3、字符串常量

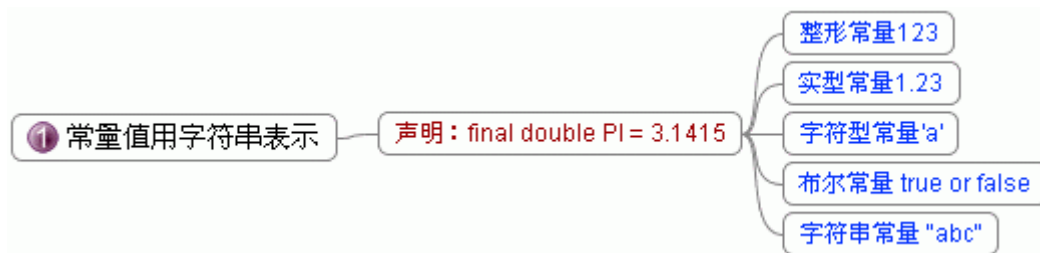
与 C、C++相同，Java 的字符串常量是用双引号(“)括起来的一串字符，如“This is a string.”。但不同的是，Java 中的字符串常量是作为 String 类的一个对象来处理的，而不是一个数据。

六、布尔型数据

布尔型数据只有两个值：true 和 false，且它们不对应于任何整数值。在流控制中常用到它。

布尔型变量的定义如：

```
boolean b = true; //定义 b 为布尔型变量，且赋值为 true。
```



(图 4-5 常量范例)

4.4 数值型数据的混合运算

1、自动类型转换

整型、实型、字符型数据可以混合运算。运算中，不同类型的数据先转化为同一类型，然后进行运算。

转换从低级到高级：byte→short→int→long→float→double、char→int

转换规则为：

- ① (byte 或 short) op int→ int
- ② (byte 或 short 或 int) op long→ long
- ③ (byte 或 short 或 int 或 long) op float→ float
- ④ (byte 或 short 或 int 或 long 或 float) op double→ double
- ⑤ char op int→ int

其中，箭头左边表示参与运算的数据类型，op 为运算符(如加、减、乘、除等)，右边表示转换成的进行运算的数据类型。

2、强制类型转换

高级数据要转换成低级数据，需用到强制类型转换，如：

```
int i;
```

```
byte b = (byte)i; //把 int 型变量 i 强制转换为 byte 型
```

这种使用可能会导致溢出或精度的下降，最好不要使用。

```
public class D10402toByte {  
    public static void main(String[] args) {  
        byte a=67;  
        byte b=89;  
        System.out.println((byte)(a+b));  
    }  
}
```

执行结果如下：

-100

具体原因请参考《从人生需要引导论 Java 快速开发》“小知识：Int 转换成 Byte 的血案”。

3、包装类过渡类型转换

在我们讨论其它变量类型之间的相互转换时，我们需要了解一下 Java 的包装类，所谓包装类，就是可以直接将简单类型的变量表示为一个类，在执行变量类型的相互转换时，我们会大量使用这些包装类。Java 共有六个包装类，分别是 Boolean、Character、Integer、Long、Float 和 Double，从字面上我们就可以看出它们分别对应于 boolean、char、int、long、float 和 double。

具体请查看 Java API 开发人员文档中 java.lang 相应类的说明。

在进行简单数据类型之间的转换（自动转换或强制转换）时，我们总是可以利用包装类进行中间过渡。

一般情况下，我们首先声明一个变量，然后生成一个对应的包装类，就可以利用包装类的各种方法进行类型转换了。例如：

当希望把 float 型转换为 double 型时：

```
float f1 = 100.00f;
Float F1 = new Float(f1);
Double d1 = F1.doubleValue();
//F1.doubleValue() 为 Float 类的返回 double 值型的方法
```

当希望把 double 型转换为 int 型时：

```
double d1 = 100.00;
Double D1 = new Double(d1);
int i1 = D1.intValue(); // 在各个包装类中，总有形为 ××Value() 的方法，来得到其对应的简单类型数据。
```

当希望把 int 型转换为 double 型时，自动转换：

```
int i1 = 200;
double d1 = i1;
```



(图 4-6 前 3 种混合运算)

4、字符串型与其它数据类型的转换

通过查阅类库中各个类提供的成员方法可以看到，几乎从 java.lang.Object 类派生的所有类提供了 toString() 方法，可将该类转换为字符串。如下所示：

```
public class D10403toString {
    public static void main(String[] args) {
        int i1 = 10;
        float f1 = 3.14f;
        double d1 = 3.1415926;
        Integer I1 = new Integer(i1); // 生成 Integer 类
        Float F1 = new Float(f1); // 生成 Float 类
        Double D1 = new Double(d1); // 生成 Double 类

        // 分别调用包装类的 toString() 方法转换为字符串
        String si1 = I1.toString();
        String sf1 = F1.toString();
        String sd1 = D1.toString();
        System.out.println("si1 " + si1);
        System.out.println("sf1 " + sf1);
        System.out.println("sd1 " + sd1);
    }
}
```

```
}  
}
```

执行结果如下：

```
si1    10  
sf1    3.14  
sd1    3.1415926
```

5、将字符型直接做为数值转换为其它数据类型

将字符型变量转换为数值型变量实际上有两种对应关系，我们在第一部分所说的那种转换中，实际上是将其转换成对应的 ASCII 码，但是我们有时还需要另一种转换关系，例如，'1' 就是指的数值 1，而不是其 ASCII 码，对于这种转换，我们可以使用 Character 的 getNumericValue(char ch) 方法。

6、Date 类与其它数据类型的相互转换

整型和 Date 类之间并不存在直接的对应关系，只是你可以使用 int 型为分别表示年、月、日、时、分、秒，这样就在两者之间建立了一个对应关系，在作这种转换时，你可以使用 Date 类构造函数的三种形式：

Date(int year, int month, int date): 以 int 型表示年、月、日

Date(int year, int month, int date, int hrs, int min): 以 int 型表示年、月、日、时、分

Date(int year, int month, int date, int hrs, int min, int sec): 以 int 型表示年、月、日、时、分、秒

在长整型和 Date 类之间有一个很有趣的对应关系，就是将一个时间表示为距离格林尼治标准时间 1970 年 1 月 1 日 0 时 0 分 0 秒的毫秒数。对于这种对应关系，Date 类也有其相应的构造函数：Date(long date)

获取 Date 类中的年、月、日、时、分、秒以及星期你可以使用 Date 类的 getYear()、getMonth()、getDate()、getHours()、getMinutes()、getSeconds()、getDay() 方法，你也可以将其理解为将 Date 类转换成 int。

而 Date 类的 getTime() 方法可以得到我们前面所说的一个时间对应的长整型数，与包装类一样，Date 类也有一个 toString() 方法可以将其转换为 String 类。

具体请参考 JDK API 中的类 java.util.Date。

有时我们希望得到 Date 的特定格式，例如 20020324，我们可以使用以下方法，首先在文件开始引入，

```
import java.text.SimpleDateFormat;
```

```
public class D10404Date {  
    public static void main(String[] args) {  
        java.util.Date date = new java.util.Date(); //如果希望得到YYYYMMDD的格式  
        SimpleDateFormat syl = new SimpleDateFormat("yyyyMMdd");  
        String dateFormat = syl.format(date);  
        System.out.println("yyyyMMdd " + dateFormat);  
    }  
}
```



```
//如果希望分开得到年，月，日
SimpleDateFormat sy = new SimpleDateFormat("yyyy");
SimpleDateFormat sm = new SimpleDateFormat("MM");
SimpleDateFormat sd = new SimpleDateFormat("dd");
String syear = sy.format(date);
String smon = sm.format(date);
String sday = sd.format(date);
System.out.println("yyyy    "+ syear);
System.out.println("MM      "+ smon);
System.out.println("dd       "+ sday);
}
}
```

执行结果如下：

```
yyyyMMdd    20090426
yyyy        2009
MM          04
dd          26
```

高级用法：

在 java 中除了这些转换之外基本数据类型还可以被隐式的转换成 String，例如：

1. 字符型转时间型：2005-10-1

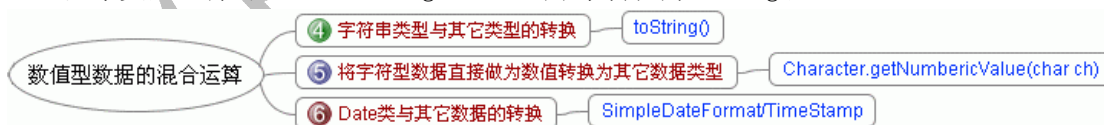
```
reportdate_str = "2005-10-01";
reportdate_str = reportdate_str + " 00:00:00.0"
reportdate = java.sql.Timestamp.valueOf(reportdate_str);
```

2. 时间型转字符型：

```
V_DATE = reportdate.toString();
```

3. 将 long 型转化为 String 型

```
long APP_UNIT = (long) userview.getDEPT_ID();
String S_APP_UNIT = String.valueOf(APP_UNIT);
基本类型 s 都可以通过 String.valueOf(s) 来转化为 String 型。
```



(图 4-7 后 3 种混合运算)

4.5 算术运算

在 Java 中提供运算功能的就是运算符 (Operator)，如与算术相关的有加(+)、减(-)、乘(*)、除(/)这类的运算符。另一个常用的是取模运算符(%)。这类以数学运算为主的运算符称为算术运算符(Arithmetic Operator)。

算术运算符的使用基本上与加减乘除一样，也是先乘除后加减，必要时加上括号表示运算的先后顺序。例如这个程序代码会在命令行模式下显示：

```
System.out.println(1 + 2 * 3);
```

编译器在读取程序代码时是由左往右读取，而初学者往往会犯一个错误，例如用 Java 表示 $(1+2+3) / 4$ 时，初学者往往写成了：

```
System.out.println(1+2+3 / 4);
```

而这个程序却是这样运算的： $1+2+(3/4)$ 。为了避免这样的错误，在必要的时候必须为表达式加上括号。例如：

```
System.out.println((double) (1+2+3) / 4);
```

注意在上面的代码中使用了 double 强制类型转换，如果不加，程序的输出结果为 1 而不是 1.5，因为 1、2、3、4 这四个数值都是整数，程序运算 $(1+2+3)$ 后的结果还是整数类型，若此时除以整数 4，会自动去掉小数点之后的数字再进行输出，而加上 double 限定，表示要将 $(1+2+3)$ 运算后的值转换为 double 数据类型，这样再除以 4，小数点之后的数字才不会被去掉。

同样地，下面代码输出 3，而不是 3.3333：

```
int testNumber = 10;
```

```
System.out.println(testNumber / 3);
```

因为 testNumber 是整数，而除数 3 也是整数，运算出来的程序被自动转换为整数了。为了解决这个问题，可以使用下面的方法：

```
int testNumber = 10;
```

```
System.out.println(testNumber / 3.0);
```

```
System.out.println((double) testNumber / 3);
```

上面这个程序片段示范了两种解决方式：如果表达式中有一个浮点数，则程序就会先转换使用浮点数来运算，这是第一段语句所使用的方式；第二种方式称为“强制类型转换”，使用 double 告诉程序先将 testNumber 的值转换为 double，然后再进行除法运算，所以得到的结果会是 3.3333。类型转换的限定关键词就是定义变量时所使用的 int、float 等关键词。

当将精确度小的数据类型(如 int)指定给精确度大的数据类型变量(如 double)时，精确度并不会失去，所以这样的指定是可行的。由于 Java 对于程序的安全性要求极高，在使用指定运算符“=”时，将精确度大的值指定给精确度小的变量时，由于在精确度上会有损失，编译器会认为这是一个错误。例如：

```
int testInteger = 0;
```

```
double testDouble = 3.14;
```

```
testInteger = testDouble;
```

```
System.out.println(testInteger);
```

这段程序在编译时会出现以下的错误信息：

```
possible loss of precision
```

```
found   : double
```

```
required: int
```

```
testInteger = testDouble
```

```
1 error
```

如果确定要转换数值为较小的精度，必须明确加上转换的限定符，编译器才不会报告错

误。如 `testInteger = (int) testDouble;`

%运算符是模运算符，它计算所得到的结果是除法后的余数。例如 `(10 % 3)` 会得到余数 1。一个使用 % 的例子是数字循环，假设有一个立方体要进行 360° 旋转，每次要在角度上加 1，而旋转 360° 后必须恢复为 0，然后重新计数。这时可以这么编写：

```
count = (count + 1) % 360;
```

4.6 比较和条件运算

数学上有比较的运算，像大于、等于、小于等运算，Java 中也提供了这些运算符，这些运算符称为“比较运算符” (Comparison Operator)。它们有大于(>)、大于等于(>=)、小于(<)、小于等于(<=)、等于(==)和不等于(!=)。

在 Java 中，比较条件成立时以 true 表示，比较条件不成立时以 false 表示。

```
public class D10405ComparisonOperator {  
    public static void main(String[] args) {  
        System.out.println("10 > 5 结果 " + (10 > 5));  
        System.out.println("10 >= 5 结果 " + (10 >= 5));  
        System.out.println("10 < 5 结果 " + (10 < 5));  
        System.out.println("10 <= 5 结果 " + (10 <= 5));  
        System.out.println("10 == 5 结果 " + (10 == 5));  
        System.out.println("10 != 5 结果 " + (10 != 5));  
    }  
}
```

执行结果略。

比较运算在使用时有一个即使是程序设计老手也可能犯的错误，且不容易发现，也就是等于运算符==。它由两个连续的等号 = 所组成，而不是一个等号，一个等号是赋值运算，这一点必须特别注意。例如，若有两个变量 x 与 y 要比较是否相等，应该是写成 `x == y`，而不是写成 `x = y`，后者的作用是将 y 的值指定给 x，而不是比较 x 与 y 是否相等。

既然谈到了条件式的问题，下面介绍 Java 中的“条件运算符”(Conditional Operator)，也叫三元运算符，它的使用方式如下：

条件式 ? 成立返回值 : 失败返回值

条件运算符的返回值依条件式的结果而定，如果条件式的结果为 true，则返回冒号前的值；若为 false，则返回冒号后的值。

若条件运算符(?:)使用得当，可以少写几句程序代码。

详情请参考“第 5 章 控制结构”中的条件控制。

4.7 逻辑和位运算

通过前面的介绍知道了大于和小于的运算，但如果想要同时进行两个以上的条件判断呢？例如分数大于 80 且小于 90 的判断。在逻辑上有所谓的与 (And)、或 (Or) 与非 (Inverse)，在 Java 中也提供这几个基本逻辑运算所需的逻辑运算符 (Logical Operator)，分别为与 (&)、或 (||) 和取反 (!) 3 个运算符。

下面来看看范例会输出什么。

```
public class D10406LogicalOperator {
    public static void main(String[] args) {
        int number = 75;
        System.out.println((number > 70 && number < 80));
        System.out.println((number > 80 || number < 75));
        System.out.println(!(number > 80 || number < 75));
    }
}
```

执行结果略。

这 3 段语句输出 true、false 与 true。

表示 number 大于 70 且小于 80 为真

number 大于 80 或小于 75 为假，

number 大于 80 或小于 75 为假，但取反后为真。

接下来看看“位运算符” (Bitwise Operator)，在数字设计上有 AND、OR、NOT、XOR 与补码等运算，在 Java 中提供这些运算的就是位运算符。它们对应的分别是 & (AND)、| (OR)、^ (XOR) 与 ~ (补码)。

如果不会基本的位运算，可从范例中了解各个位运算的结果。

```
public class D10407BitwiseOperator {
    public static void main(String[] args) {
        System.out.println("AND运算: ");
        System.out.println("0 AND 0\t\t" + (0 & 0));
        System.out.println("0 AND 1\t\t" + (0 & 1));
        System.out.println("1 AND 0\t\t" + (1 & 0));
        System.out.println("1 AND 1\t\t" + (1 & 1));

        System.out.println("\nOR运算: ");
        System.out.println("0 OR 0\t\t" + (0 | 0));
        System.out.println("0 OR 1\t\t" + (0 | 1));
        System.out.println("1 OR 0\t\t" + (1 | 0));
        System.out.println("1 OR 1\t\t" + (1 | 1));

        System.out.println("\nXOR运算: ");
        System.out.println("0 XOR 0\t\t" + (0 ^ 0));
        System.out.println("0 XOR 1\t\t" + (0 ^ 1));
        System.out.println("1 XOR 0\t\t" + (1 ^ 0));
        System.out.println("1 XOR 1\t\t" + (1 ^ 1));
    }
}
```



```
}  
}
```

执行结果略。

Java 中的位运算是逐位运算的，例如 10010001 与 01000001 做 AND 运算，是一个一个位对应运算，答案就是 00000001；而补码运算是将所有的位 0 变 1，1 变 0，例如 00000001 经补码运算就会变为 11111110。例如下面这个程序所示：

```
public class D10408Bitwise {  
    public static void main(String args[]) {  
        byte number = 0;  
        System.out.println((int) (~number));  
  
        number = -1;  
        printBinaryInt(~number);  
    }  
  
    static void printBinaryInt(int i) {  
        System.out.print("int:" + i + ",binary:");  
        for (int j = 31; j >= 0; j--)  
            if (((1 << j) & i) != 0)  
                System.out.print("1");  
            else  
                System.out.print("0");  
        System.out.println();  
    }  
}
```

执行结果如下：

```
-1  
int:0,binary:00000000000000000000000000000000
```

因为 byte 占内存一个字节，它存储的 0 在内存中是 00000000，经补码运算就变成 11111111，这个数在计算机中用整数表示则是 -1。

补码的详细内容请参考“4.9 计算机基础：原码、反码、补码”。

要注意的是，逻辑运算符与位运算符经常被混淆的，像&&与&、||与|，初学时可得多注意。位运算对初学者来说的确不常用，但如果用得恰当，可以提高程序的运算效率。例如下面的代码可以判断使用者的输入是否为奇数：

```
import java.util.Scanner;  
  
public class D10409OddDecider {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);
```

```
System.out.print("请输入数字: ");  
int number = scanner.nextInt();  
System.out.println("是否为奇数? " +  
    ((number&1) != 0 ? '是' : '否'));  
}  
}
```

执行结果如下:

请输入数字: 70

是否为奇数? 否

以上代码的原理是，奇数的数值若以二进制来表示，其最右边的位必为 1，而偶数最右边的位必为 0，所以使用 1 与输入的值做 AND 运算。由于 1 除了最右边的位为 1 之外，其他位都会是 0，与输入数值 AND 运算的结果，只会留下最右边位为 0 或为 1 的结果，其他部分都被 0 AND 运算遮掉了，这就是所谓的“位屏蔽”。例如 4 与 1 做 AND 运算的结果会是 0，所以判断为偶数：

整数 4: 00000100

整数 1: 00000001

AND 运算后: 00000000

而 3 与 1 做 AND 运算的结果是 1，所以判断为奇数：

整数 3: 00000011

整数 1: 00000001

AND 运算后: 00000001

XOR 的运算不常见，下面是一个简单的 XOR 字符加密例子。

```
public class D10410XorCode {  
    public static void main(String[] args) {  
        char ch = 'A';  
        System.out.println("编码前: " + ch);  
  
        ch = (char) (ch^7);  
        System.out.println("编码后: " + ch);  
  
        ch = (char) (ch^7);  
        System.out.println("解码: " + ch);  
    }  
}
```

执行结果如下:

编码前: A

编码后: F

解码: A

0x7 是 Java 中整数的十六进制写法，其实就是十进制的 7，将位与 1 作 XOR 的作用其实就是位反转，0x7 的最右边 3 个位为 1，所以就是反转 ch 变量的最后两个位，如下所示：

ASCII 中的 'A' 字符编码为 65：01000001

整数 7：00000111

XOR 运算后：01000110

01000110 就是整数 70，对应 ASCII 中的字符 F 的编码，所以用字符方式显示时会显示 F 字符。同样地，这个简单的 XOR 字符加密，要解密也只要再进行相同的位反转就可以了。

注意的是，以上的说明都只写下 8 位，这只是为了解说方便。实际的位长度在运算时，需依数据类型所占的内存长度而定。例如在使用 int 类型的 0 做运算时，要考虑的是 32 位长度，而不是只有 8 位，因为 int 占有 4 个字节，也就是实际上是 00000000 00000000 00000000 00000000。

在位运算上，Java 还有左移(<<)与右移(>>)两个运算符。左移运算会将所有的位向左移指定的位数，左边被挤出去的位会被丢弃，而右边会补上 0；右移运算则相反，会将所有的位向右移指定的位数，右边被挤出去的位会被丢弃，最左边补上原来的位，即左边原来是 0 就补 0，原来是 1 就补 1。

下面代码使用左移运算来做简单的 2 次方运算示范。

```
public class D10411ShiftOperator {
    public static void main(String[] args) {
        int number = 1;
        System.out.println("2的0次方: " + number);

        number = number << 1;
        System.out.println("2的1次方: " + number);

        number = number << 1;
        System.out.println("2的2次方: " + number);

        number = number << 1;
        System.out.println("2的3次方: " + number);
    }
}
```

执行结果略。

注意：位运算对于没有学过数字逻辑的人来说，难以理解，基本上除了像信息工程、电机工程相关领域的开发人员会比较常使用位运算之外，大部分的开发人员很少使用位运算。

4.8 递增和递减与赋值运算

递增(Increment)、递减(Decrement) (Assignment) 运算符，老实说常成为初学者的一个恶梦，因为有些程序若写得精简，这几个运算符会让初学者搞不清楚程序的真正运算结果是

什么。事实上，使用这几种运算符的目的除了是让程序看起来比较简洁之外，还可以稍微增加一些程序执行的效率。

在程序中对变量递增 1 或递减 1 是很常见的运算，例如：

```
int i = 0;
i = i + 1;
System.out.println(i);
i = i - 1;
System.out.println(i);
```

这段程序会分别显示出 1 与 0 两个数，如使用递增、递减运算符来编写，代码如下：

```
public class D10412IncrementDecrement {
    public static void main(String[] args) {
        int i = 0;
        System.out.println(++i);
        System.out.println(--i);
    }
}
```

执行结果略。

其中写在变量 `i` 之前的 `++` 与 `--` 就是“递增运算符” (Increment Operator) 与“递减运算符” (Decrement Operator)，当它们出现在变量之前时，其作用就相当于将变量递增 1 与递减 1：

```
++i;    // 相当于 i = i + 1;
--i;    // 相当于 i = i - 1;
```

可以将递增或递减运算符编写在变量之前或变量之后，但这两种写法是有差别的。将递增(递减)运算符编写在变量前时，表示先将变量的值加(减)1，然后再返回变量的值，将递增(递减)运算符编写在变量之后，表示先返回变量值，然后再对变量加(减)1。例如：

```
public class D10413IncrementDecrement2 {
    public static void main(String[] args) {
        int i = 0;
        int number = 0;

        number = ++i;    // 相当于 i = i + 1; number = i;
        System.out.println(number);
        number = --i;    // 相当于 i = i - 1; number = i;
        System.out.println(number);
    }
}
```

执行结果略。

在这段程序中，`number` 的值会前后分别显示为 1 与 0，再看看以下代码。

```
public class D10414IncrementDecrement3 {
    public static void main(String[] args) {
```

```
int i = 0;
int number = 0;

number = i++;    // 相当于 number = i; i = i + 1;
System.out.println(number);
number = i--;    // 相当于 number = i; i = i - 1;
System.out.println(number);
}
}
```

执行结果略。

在这段程序中，number 的值前后分别显示为 0 与 1。

接下来看“赋值运算符” (Assignment Operator)。到目前为止您只看过一个赋值运算符，也就是=这个运算符，事实上赋值运算符还有其他几个，如下表所示。

赋值运算符	范 例	结 果
+=	a += b	a = a + b
-=	a -= b	a = a - b
*=	a *= b	a = a * b
/=	a /= b	a = a / b
%=	a %= b	a = a % b
&=	a &= b	a = a & b
=	a = b	a = a b
^=	a ^= b	a = a ^ b
<<=	a <<= b	a = a << b
>>=	a >>= b	a = a >> b

每个赋值运算符的作用如上所示，但老实说若不是常写程序的老手，当遇到这些赋值运算符时，有时可能会愣一下。因为不常用的话，这些语法并不是那么的直观。

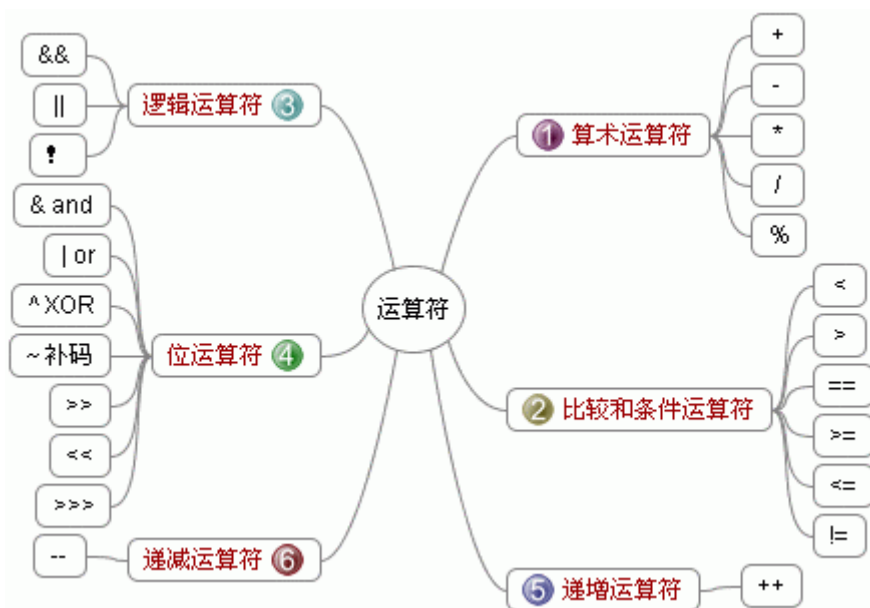
使用++、--或赋值运算符，由于程序可以直接在变量的内存空间中运算，而不用取出变量值、运算再将数值存回变量的内存空间，所以可以增加运算的效率。但以现在计算机的运算速度来看，这一点的效率可能有些微不足道，除非这类的运算相当的频繁，否则看不出这点效率所带来的改善。就现在程序编写的规模来看，程序的易懂、易读、易维护会比效率更重要。可以的话尽量将程序写得详细一些会比较好，千万不要为了卖弄语法而滥用这些运算符。

单就一个语句而言，使用++、--或赋值运算符还可以理解，但与其他语句结合时可就应考虑一下。例如：

```
int i = 5;
arr[--i %= 10] = 10;
```

像这样的式子，要想知道变量 i 是多少，以及数组的指定索引位置在哪儿可就得想一下了(有兴趣算一下的话，i 最后会是 4，而数组的指定索引也是 4)。总之，如何使用与何时

使用，自己得揣摩着点。



(图 4-8 运算)

4.9 计算机基础：原码、反码、补码

原码是机器数的一种简单的表示法。数值在计算机中表示形式为机器数，计算机只能识别 0 和 1，使用二进制，但日常生活中使用十进制。数值有正负之分，计算机用一个数的最高位存放符号(0 为正，1 为负)。

假设机器能处理的位数为 8，即字长为 1 字节，原码能表示数值的范围为 $(-127 \sim -0 + 0 \sim 127)$ 共 256 个。

有了数值的表示方法就可以对数进行算术运算，当两数相加时，如果同号则数值相加；如果异号，则要进行减法。并且在进行减法时需要比较绝对值的大小，然后大数减去小数，最后还要给结果选择符号。直接运用这种方式去处理，计算机会很麻烦。

为了解决这些矛盾，人们找到了补码表示法，即数值一律用补码来表示（存储）。机器数的补码可由原码得到。如果机器数是正数，则该机器数的补码与原码一样；如果机器数是负数，则该机器数的补码是对它的原码（除符号位外）各位取反，并在末位加 1 而得到。

在补码中用 (-128) 代替了 (-0) ，所以补码的表示范围为： $(-128 \sim 0 \sim 127)$ 共 256 个。

注意： (-128) 没有相对应的原码和反码， $(-128) = (10000000)$ 。

对除符号位外的其余各位逐位取反就产生了反码（对于正数，其反码与原码相同）。反码的取值空间和原码相同且一一对应。

负数用补码表示时，可以把减法转化为加法。这样，在计算机中实现起来就很方便。

补码的原理可以用钟表来描述：如设标准时间为 4 点正；一只表已经 7 点了，为了校准时间，可以采用两种方法：一是将时针退 $7-4=3$ 格；一是将时针向前拨 $12-3=9$ 格。即 $7-3$ 和 $7+9 \pmod{12}$ 等价，因此，把负数用补码表示的 $\pmod{2}$ 操作，可以把减法转化为加法。使用补码，可以将符号位和其它位统一处理；同时，减法也可按加法来处理。另外，两个用补码表示的数相加时，如果最高位（符号位）有进位，则进位被舍弃。

所以补码的设计目的是：

- (1)使符号位能与有效值部分一起参加运算，从而简化运算规则。
- (2)使减法运算转换为加法运算，进一步简化计算机中运算器的线路设计。

二进制补码定义形式非常漂亮，因为它具有对称性。在二进制里面，很多的东西都是“恰好如此”。恰好二进制只有 0、1 两个数，恰好只有正负两种状态，恰好 $2^n + 2^n = 2^{n+1}$ ……在补码的对应规则下，n 个数位能表示的数的个数还是那么多，只不过把其中的一半的正数的形式让给了负数。

正数：补码、反码、原码相同。

负数：补码是正数取反加一。

最高位为符号位：0 代表正，1 代表负。

1 的 16 位编码 0000000000000001

-1 的编码为 1 的编码取反+1：

1 的编码取反为：1111111111111110

然后再加 1 为：1111111111111111

这就是-1 的编码。

32767 的编码 0111111111111111

取反为 1000000000000000

再加 1 得到-32767 的编码：

1000000000000001

-32768 比 32767 还少 1，计算如下：

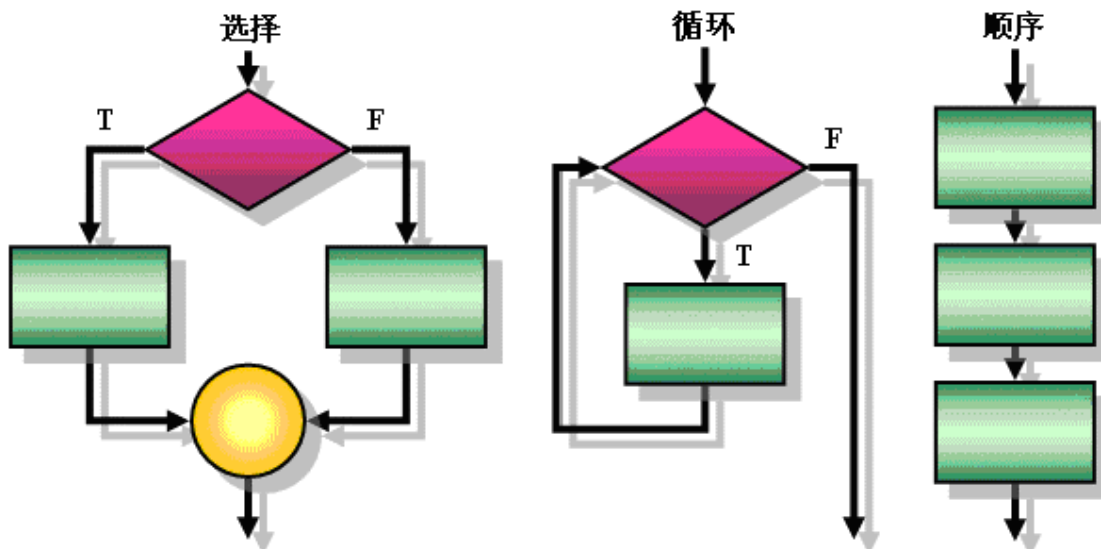
1000000000000001

-0000000000000001

= 1000000000000000

第 5 章 控制结构

根据结构定理(structure theorem)，任何计算机程序都可以用下图中的基本控制结构来表示。它们可以任意组合来解决问题。

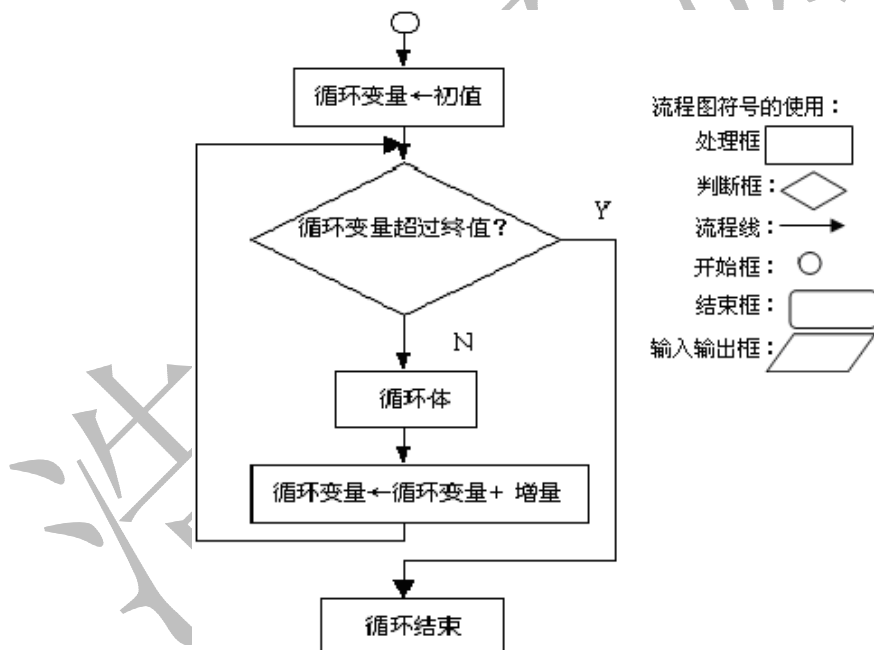


(图 5-1 基本控制结构)

选择结构 (selection structure)，用于测试条件，根据条件的真假，执行一系列语句。一个条件语句可以是任何能够返回布尔值 (TRUE 或 FALSE) 的变量或表达式。

循环结构 (repetition structure)，能在条件满足的情况下反复执行。

顺序结构 (Sequence structure)，只是简单的按照顺序执行语句。



(图 5-2 程序流程图)

如图 5-2 所示，程序流程图就是控制结构的组合，但目前已经很少使用。

5.1 条件控制

条件控制有两种：IF 和 SWITCH 语句

有时候，我们需要根据具体的条件来采取不同的对策。IF 语句就能让我们按条件来执行语句序列。也就是说，语句序列的执行与否取决于某个给定的条件。

有三种 IF 语句：IF（单选）、IF-ELSE（多选）和 IF-ELSE IF（多重选择）。

SWITCH 语句是条件判断的精简形式，它能计算条件表达式的值并在多个对应动作中做出选择。

1、IF（单选）

IF 语句最简单的形式就是把一个条件和一个语句，例：

if（表达式）

语句 1

“表达式”应当为算术类型或者指针类型。“语句 1”的代码缩进既强调了结构化程序设计的继承结构，又可以程序条理清晰，给人舒服的感觉。

例如有以下伪码：

```
if student's grade is greater than or equal to 60
    print "Passed"
```

上述伪码变换为 Java 语句则是：

```
if(studentGrade>=60)
    System.out.println("Pass");
```

由此可看出伪码和实际的 java 语句十分接近，使程序员的条理更加清晰。

几个语句组合在一起组成复合语句，在形式上是语句前后用一对花括号：{} 括起来，其中可以有一系列的定义或说明，后跟一系列的语句。如：

if（表达式）

```
{
    语句 1;
    语句 2;
    ...
    语句 N
}
```

2、IF-ELSE（多选）

使用关键字 ELSE 添加了一个额外的处理选项，如下：

if（表达式）语句 1 else 语句 2

当条件为假或空时，ELSE 子句中的语句 2 就会被执行。

当然 IF-ELSE 的子句为复合语句时，中间可以包含 IF 语句。就是说 IF 语句能够被嵌套使用。例：

if（表达式）

```
{
    语句 1
    if（表达式 1）
        语句 01
}
```

```
else  
    语句 2
```

为了养成良好的编码风格，建议即使只有一个语句，也要用一对花括号：{}括起来。

```
if (表达式) {语句 1}  
else {语句 2}
```

例如伪码：

```
if student grade is greater than or equal to 60  
    print"passed"  
else  
    print"failed"
```

转换为 java 语句则是：

```
if(studentGrade>=60) {System.out.println("passed");}  
else {System.out.println("Failed");}
```

Java 还提出了唯一的三元运算符 "?:", 则上述的 java 程序可写为：
System.out.println(studentGrade>=60?"passed":"Failed");

用 Java 术语来说，三元运算符，是“条件运算符” (Conditional Operator)，使用方式如下：

条件式 ? 成立返回值 : 失败返回值

条件运算符的返回值依条件式的结果而定，如果条件式的结果为 true，则返回冒号前的值；若为 false，则返回冒号后的值。

若条件运算符(?:)使用得当，可以少写几句程序代码。即这样可以使程序长度缩短，缩短运行时间。

```
public class D10501Grade {  
    public static void main(String[] args) {  
        int studentGrade=71;  
  
        if(studentGrade>=60) {System.out.println("passed");}  
        else {System.out.println("Failed");}  
  
        //三元运算符  
        System.out.println(studentGrade>=60?"passed":"Failed");  
    }  
}
```

执行结果略。

3、IF-ELSE IF（多重选择）

如需要从几个选项中选择一个时，我们将 IF 语句组合起来使用。如果第一个条件为假或空，ELSE IF 子句就会检测另外一个条件。一个 IF 语句可以有多个 ELSIF 子句；最后一个 ELSE 子句是可选的。条件表达式从上而下的计算。只要有满足的条件，与它关联的语句

才会执行，然后控制权转到下一个语句。如果所有的条件都为假或是空，ELSE 部分的语句就会执行。

4、SWITCH（多重选择）：

SWITCH 是控制选择的一种方式，编译器生成代码时可以对这种结构进行特定的优化，从而产生效率比较高的代码。

```
switch (表达式) {case 常量: 语句序列 case 常量: 语句序列 ...}
```

同 IF 语句一样，SWITCH 语句也是选出一个语句序列来执行。但是，为了选择出合适的语句序列，SWITCH 会使用一个选择器，而不是多个布尔表达式。

这里“表达式”和“常量”可以是各种字符类型或整型。首先求“表达式”的值，然后用这个值与作为 SWITCH 标号的各常量匹配。如果发现匹配的标号，就从这个位置开始执行。

一个 SWITCH 语句中的标号不能有相同的值。语句中可以有一个 default 标号，其他标号都不匹配时执行转到 default 标号处。如果没有 default 标号，出现标号都不匹配的情况时语句立即结束。人们通常在每个序列的最后放一个 break 语句。范例：

```
public class D10502Switch {  
    public static void main(String[] args) {  
        int i = 3;  
        switch (i) {  
            case 0:  
                System.out.println("0");break;  
            case 1:  
                System.out.println("1");break;  
            case 3:  
                System.out.println("3");break;  
            case 5:  
                System.out.println("5");break;  
            case 10:  
                System.out.println("10");break;  
            case 13:  
                System.out.println("13");break;  
            case 14:  
                System.out.println("14");break;  
            default:  
                System.out.println("default");break;  
        }  
    }  
}
```

执行结果略。

SWITCH 语句的可读性高而且高效，所以，如果可能的话，尽量把 IF-THEN-ELSEIF 都改写成 SWITCH 语句。

5.2 循环控制

1、for 循环

如果要进行重复性的指令执行，可以使用 for 循环式。它的基本语法如下：

```
for(初始式; 判断式; 递增式) {  
    语句一;  
    语句二;  
}
```

如果语句只有一个，也就是非复合语句，{ } 可以省略不写。for 循环的第一个初始语句只会执行一次，之后每次重新进行循环时，都会根据判断式来判断是否执行下一个循环，而每次执行完循环之后，都会执行递增式。

下面得代码可以了解 for 循环的功能。

```
public class D10503SimpleLoopFor {  
    public static void main(String[] args) {  
        for(int i = 0; i < 10; i++)  
            System.out.print(" " + i);  
    }  
}
```

执行结果如下： 0 1 2 3 4 5 6 7 8 9

这是一个简单的例子，但说明 for 的作用再适合不过。在 Java 中可以直接在 for 中定义变量与指定初始值，这个定义的变量在 for 循环结束之后也会自动消失。初始变量的语句只被执行一次，接下来循环会根据 i 是否小于 10 来判断是否执行循环，而每执行一次循环就将 i 加 1。

for 循环中也可以再使用 for 循环。

事实上，for 循环的语法只是将 3 个复合语句程序块写在一个括号中而已，所不同的是第一个语句程序块只会执行一次，第二个语句程序块专门判断是否继续下一个循环，而第三个语句程序块只是一般的语句，而不一定只放递增表达式。

for 括号中的每个语句程序块是以分号;作区隔，而在一个语句程序块中若想写两个以上的语句，则使用逗号,作区隔。下面的代码展示九九表的打印。

```
public class D10504NineTable {  
    public static void main(String[] args) {  
        int i, j;  
        for (i=1;i<10;i++) {  
            for (j=1;j<=i;j++) {  
                System.out.printf("%d*%d=%2d%c", j, i, i * j, (i == j ? '\n' : ' '));  
            }  
        }  
    }  
}
```

//两重循环合并

```
for (i = 1, j = 1; i<10&& j<=i; j=(j++==i)?((i++/i)+1):j) {  
    System.out.printf("%d*%d=%2d%c", j, i, i * j, (i == j ? '\n' : ' '));  
}
```

```
}  
}  
  
}
```

执行结果略。

这个程序的重点在于让你看看 for 循环的写法。但在原则上不鼓励这么写，程序要以清楚易懂为原则。

2、while 循环

while 循环式，根据指定的条件式来判断是否执行循环体。语法如下：

```
while(条件式) {  
    语句一;  
    语句二;  
}
```

如果循环体只有一个语句，则 while 的 { } 可以省略不写。While 等同于没有起始语句与终止语句的 for 循环，主要用于重复性的动作，而停止的时机是未知的。

while 可以用作无穷循环，很多地方都用到无穷循环。例如游戏设计中对使用者输入设备的轮询(Poll)，或是动画程序的播放都会使用到无穷循环。一个无穷循环如下所示：

```
while(true) {  
    循环内容;  
    ...  
}
```

无穷循环可以由自己循环中的某个条件式来结束。下面是一个循环内部终止的例子：

```
while(true) {  
    语句;  
    if(条件式)  
        break; // 跳离循环  
    ...  
}
```

当条件式成立时，会执行 break 离开 while 循环，这个 break 与 switch 中的作用是一样的，都是要离开当时执行的程序块时使用。

while 循环有时称为“当型循环”，因为它在循环执行前就会进行条件判断，而另一个 do while 称为“直到型循环”，它会先执行循环体，然后再进行条件判断。do while 的语法如下所示：

```
do {  
    语句一;  
    语句二;  
    ...  
} while(条件式);
```

注意 while 后面是以分号;作为结束，这个经常被忽略。由于 do while 会先执行循环，所以它通常用于进行一些初始化或接口沟通的动作。

```
public class D10505While {  
    public static void main(String[] args) {
```

```
int i=0;
//当型循环
System.out.print("当型循环: ");
while (i<10){
    System.out.print(" " + i++);
}

//直到型循环
System.out.print("\n直到型循环(当前i值为"+i+"): ");
do{
    System.out.print(" " + i++);
}while (i<10);    //必须以分号;结束

//无穷循环
System.out.print("\n无穷循环: ");
while(true){
    System.out.print(" " + i++);
    if (i>15)
        break;
}
}
```

执行结果略。

3、break 和 continue

break 可以离开当前 switch、for、while、do while 的程序块，并前进至程序块后下一条语句，在 switch 中主要用来中断下一个 case 的比较。在 for、while 与 do while 中，主要用于中断目前的循环执行。

continue 的作用与 break 类似，主要用于循环，所不同的是 break 会结束程序块的执行，而 continue 只会结束其之后程序块的语句，并跳回循环程序块的开头继续下一个循环，而不是离开循环。例如：

```
public class D10506Break {
    public static void main(String[] args) {
        for(int i = 1; i < 10; i++) {
            if(i == 5)
                break;

            System.out.print(" " + i);
        }
    }
}
```

执行结果略。

这段程序会显示 i = 1 到 4，因为当 i 等于 5 时就会执行 break 而离开循环。再看下

面这个程序：

```
public class D10507Continue {  
    public static void main(String[] args) {  
        for(int i = 1; i < 10; i++) {  
            if(i == 5)  
                continue;  
  
            System.out.print(" " + i);  
        }  
    }  
}
```

执行结果略。

这段程序会显示 1 到 4，与 6 到 9。当 i 等于 5 时，会执行 continue 直接结束此次循环，这次循环中 System.out.print() 行并没有被执行，然后从程序块开头执行下一次循环，所以 5 并没有被显示。

break 与 continue 还可以配合标签使用，例如本来 break 只会离开 for 循环，设定标签与程序块，则可以离开整个程序块。下面是一个简单的示范。

```
public class D10508BreakTest {  
    public static void main(String[] args) {  
        back: {  
            for (int i = 0; i < 10; i++) {  
                if (i == 9) {  
                    System.out.println("break");  
                    break back;  
                }  
            }  
            System.out.println("test");  
        }  
    }  
}
```

执行结果如下：break

程序的执行结果会显示 break；back 是个标签，当 break back; 时，返回至 back 标签处，之后整个 back 程序块不执行而跳过，所以这个程序 System.out.println("test"); 不会被执行。

事实上 continue 也有类似的用法，只不过标签只能设定在 for 之前。下面代码是一个简单的示范。

```
public class D10509ContinueTest {  
    public static void main(String[] args) {  
        back1: for (int i = 0; i < 10; i++) {  
            back2: for (int j = 0; j < 10; j++) {  
                if (j == 9) {  
                    continue back1;  
                }  
            }  
        }  
    }  
}
```



```
    }  
    }  
    System.out.println("test");  
    }  
}
```

执行结果如下：（无内容）

continue 配合标签，可以自由地跳至任何一层 for 循环，你可以试试 continue back1 与 continue back2 的不同。设定 back1 时，System.out.println("test"); 不会被执行。

4、return

return 返回语句有两种方式：

return;

return 表达式;

立即结束当前函数的执行，控制返回函数调用处。第一种形式用于无返回值的函数，第二种形式用于有返回值的函数。第二种形式在执行时表达式被求值，得到的值转换成函数所要求的返回值类型，作为函数的返回值。

第 6 章 基于数据库的 Hello World

“Hello World”作为开发的入门篇，已成为习惯。数据库开发的 Hello World 是怎样的呢？通过本章的学习，能初步掌握数据库开发的基本知识。

数据操作是一个过程，进行具体操作前，先申请数据库资源，操作完成后，必须释放资源。常用的操作主要针对记录，分为增删改查-CRUD 四种。

CRUD 是一个数据库技术中的缩写词，一般项目开发中各种数据的基本操作都是 CRUD。它代表创建（Create）、读取（Read）、更新（Update）和删除（Delete）操作。

CRUD 定义了用于处理数据的基本原子操作。

之所以将 CRUD 提升到一个技术难题的高度，是因为完成一个对多个数据库系统同时进行 CRUD 操作的功能时，其性能可能会随数据关系的变化而有非常大的差异。

在开始数据库的学习之前，先阐述一个观点。我们大都已经使用电脑 N 久了，然而对操作系统，有几个人真正了解？绝大多数人只会使用，甚至还只会用极小的一部分功能，至于操作系统的原理就更不用说了。我们有必要去了解那么多吗，操作系统只是一个工具而已，使用其中的部分功能就可以给我们带来方便。

对于学习，有些东西也不必去了解很深。因为有些东西，如果真要去说个所以然呢，还真难。比如本章中说到的申请数据库资源，为啥要那样写？没法说得很明白。要说呢，只能说这是一个约定俗成的规定，当初就是这样定义的。所以很多东西，我们照着做就行，尤其是初学者，一定要先当一个严格的执行者，照着去做，方能快速入门。等你入门了，就可以去了解更多的东西，否则你只是浪费时间、虚度光阴。

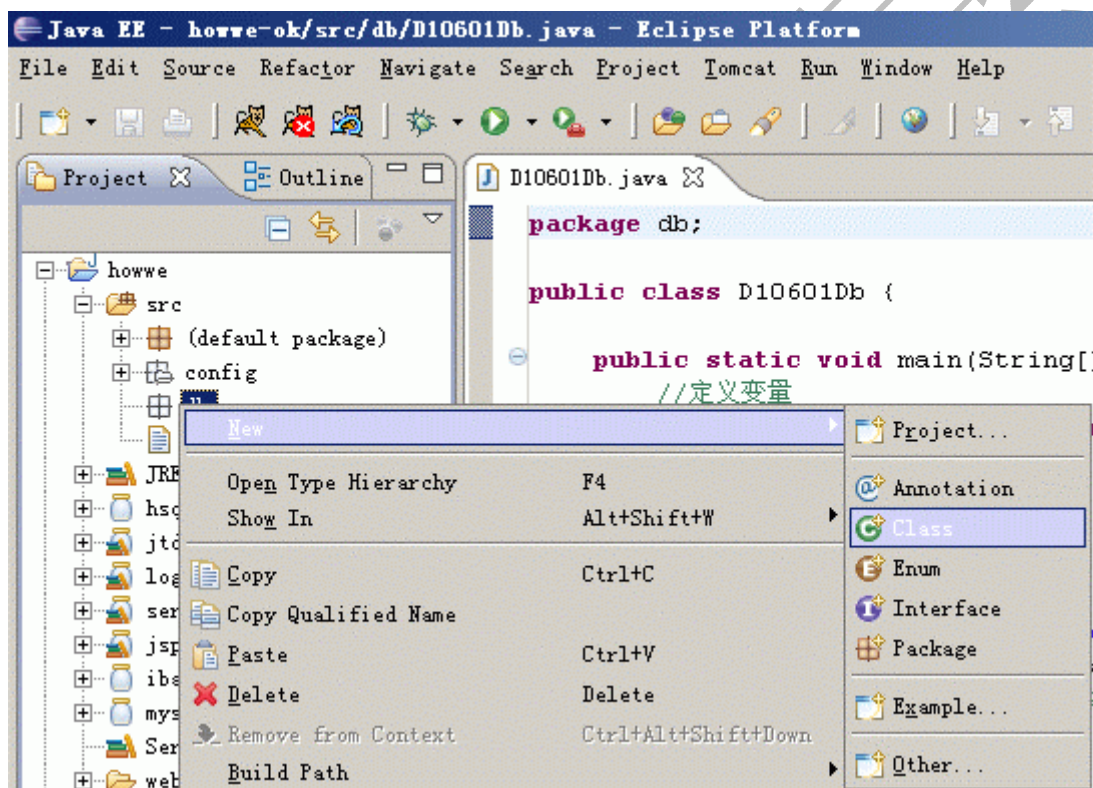
当然，如果你很牛很有能力，你完全可以自己设计一整套东西，让别人来用你的东西。但别人为啥要用你的东西呢，你就得好好去思考咯。

6.1 数据操作种类

常用的数据库大部分需要安装, HSQL 是一款纯 Java 编写的免费数据库, 不用安装。HSQL 体积小, 一个 hsqldb.jar 文件, 仅有 675kb (版本 1.8.0), 就已经包括数据库引擎、数据库驱动、用户操作界面等。详情请参考“第 8 章 认识开源数据 HSQLDB”。

下载地址: <http://hsqldb.org/>, 但范例中已包含。

先看一个例子: D10601Db.java



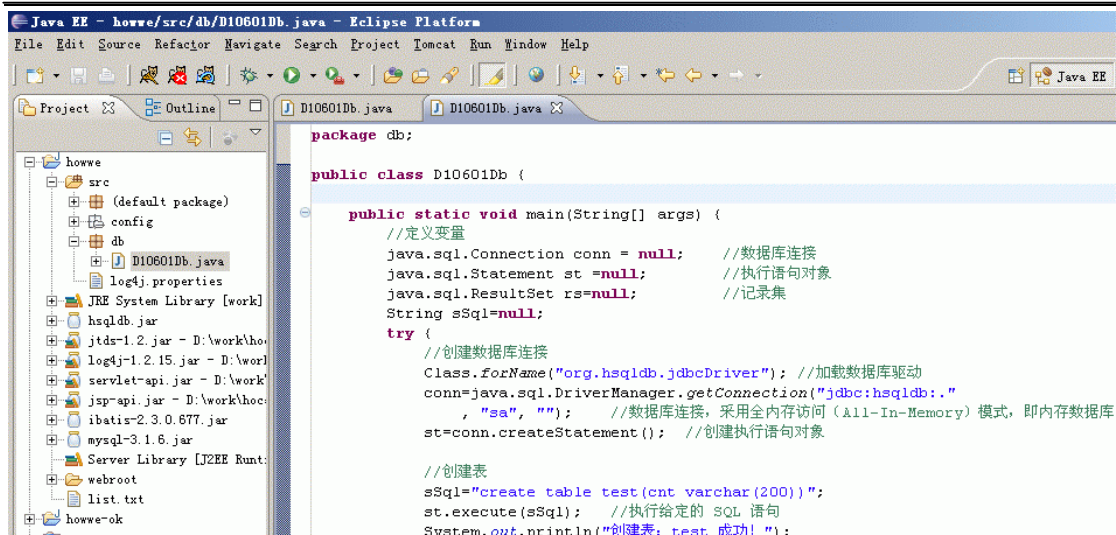
(图 6-1 创建 D10601Db 类)

参照图 6-1, 先在 src 目录上单击右键, 然后选择 **New => Package**。在 Name 处键入 db 作为包的名称, 然后按 **Finish**。

在包 db 上单击右键, 然后选择 **New => Class**。在 Name 处键入 D10601Db 作为类的名称。在“Which method stubs would you like to create?”下面, 选中“public static void main(String[] args)”复选框, 然后按 **Finish**, 操作详情请参考 3.1 节。

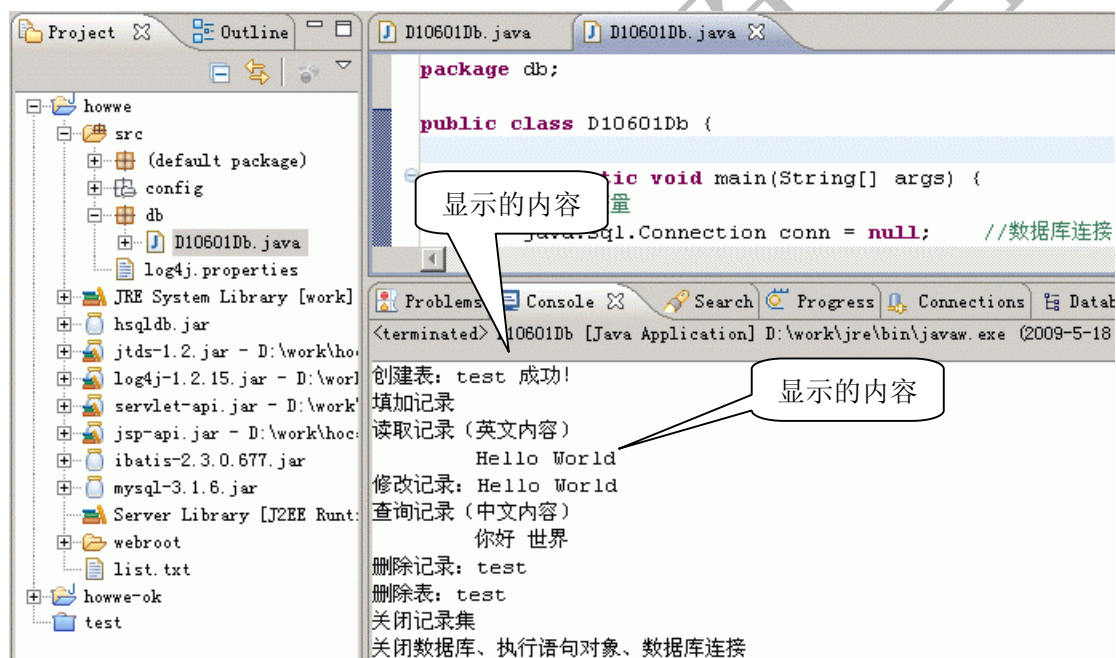
在 main() 方法里加入以下代码:

为了缩减篇幅, 代码略, 后面将分段讲解, 请自己查看 howwe-ok 项目 db/D10601Db.java, 文件具体路径: D:\work\hocat\hocas-ok\src\db\D10601Db.java。



(图 6-2 创建 D10601Db 部分代码)

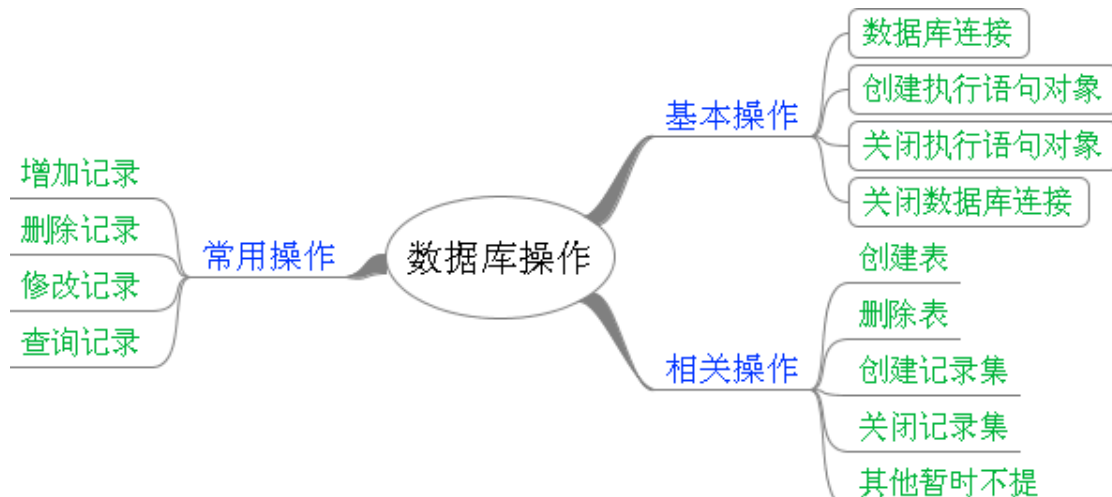
运行结果如图 6-3 所示，显示的内容见图的提示。



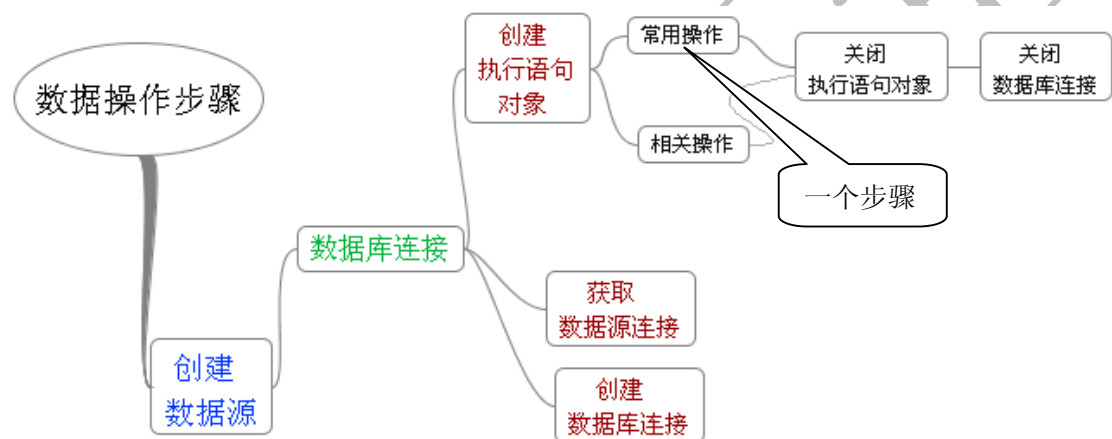
(图 6-3 D10601Db 运行结果)

初学者一看这么多代码，估计会头大。但是就这点代码，就完成了创建数据库连接、创建表、增加记录、删除记录、修改记录、查询记录、删除表、关闭记录集、关闭执行语句对象、关闭数据库、关闭数据库连接等与数据库相关的所有操作。

数据库常用操作有增加记录、删除记录、修改记录、查询记录，创建数据库连接、创建执行语句对象、关闭执行语句对象、关闭数据库连接为基本操作，其余的操作为常用操作所需的相关操作。分类为图 6-4 所示。



(图 6-4 数据库操作分类)



(图 6-5 数据操作过程)

数据操作过程如图 6-5 所示，常用操作和相关操作只是数据操作中的一个步骤。数据库连接是第一个步骤；创建执行语句对象是第二个步骤；具体操作是第三个步骤，由常用操作或相关操作的一个或多个构成；操作结束，先关闭执行语句对象、再关闭数据库连接。

其中数据库连接有两种方式，一种是前面例子所用的创建数据库连接，另一种是获取数据源连接。



(图 6-6 数据操作过程具体步骤)

如图 6-6 所示，在进行数据库的具体操作前，先准备好操作的条件，具体操作完成后，还得恢复原来的状态，所以要关闭执行语句对象、关闭数据库连接。这样才能使操作能持续的进行，而不至于耗尽计算机的资源。

通俗一点讲，不论对计算机进行何种操作，我们先要准备资源，操作完成后，必须释放资源。如果不释放，资源只会越用越少，以致资源耗尽。

日常使用的页面中，一般只有一至两种操作，所以单纯写点数据库操作方面的代码很容易，唯一难的是要能整体掌握项目的需求。故浩为一直看重整体思维，而不只是一个一个的知识点。而这些知识点，用 Google 一搜一大把，同时浩为也将整理一系列的知识点，发布

在网站 howwe.net 上。

所以本书不会讲述很多详细的知识点，只会提出一些能将这些知识点串起来的过程，具体的知识点请自己去搜索。

6.2 申请资源

申请资源，是进行具体数据操作的前提。主要步骤：

- 1、建立数据库连接
- 2、创建执行语句对象。

代码如下：

```
//定义变量
java.sql.Connection conn = null;    //数据库连接
java.sql.Statement st = null;       //执行语句对象
try {
    //创建数据库连接
    Class.forName("org.hsqldb.jdbcDriver");    //加载数据库驱动
    conn=java.sql.DriverManager.getConnection("jdbc:hsqldb:."
        , "sa", "");    //数据库连接，采用全内存访问（All-In-Memory）模式，
    即内存数据库
    st=conn.createStatement();    //创建执行语句对象

} catch (java.sql.SQLException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
```

主要采用 java.sql 提供的方法。java.sql 是提供使用 Java™ 编程语言访问并处理存储在数据源（通常是一个关系数据库）中的数据的 API。其中常用的如下：

DriverManager 管理一组 JDBC 驱动程序的基本服务。

Connection 与特定数据库的连接（会话）。

Statement 用于执行静态 SQL 语句并返回它所生成结果的对象。

PreparedStatement 表示预编译的 SQL 语句的对象。

CallableStatement 用于执行 SQL 存储过程的接口。

后两种在以后会用到。DriverManager 是一种最基础的驱动管理方式，目前大多数系统采用数据库连接池技术。

6.3 常用数据操作

常用数据操作，主要用于操作记录，简单一点说，就四个字：增删改查-CRUD。

增：创建/增加，Create/insert

删：删除，Delete

改：修改，Update

查：读取/查询，Read/select

CRUD是一个数据库技术中的缩写词，一般项目开发的各种数据的基本功能都是CURD。它代表创建（Create）、读取（Read）、更新（Update）和删除（Delete）操作。

CRUD 定义了用于处理数据的基本原子操作。

之所以将CRUD提升到一个技术难题的高度，是因为完成一个对多个数据库系统同时进行CRUD操作的功能时，其性能可能会随数据关系的变化而有非常大的差异。

```
//增加记录
sSql="insert into test values('Hello World')";
st.execute(sSql);
System.out.println("添加记录");

//查询记录
sSql="SELECT cnt FROM test";
rs = st.executeQuery(sSql);
System.out.println("读取记录（英文内容）");
while(rs.next()) {
    System.out.println("\t"+rs.getString(1));
}

//修改记录
st.execute("update test set cnt='你好 世界' where cnt='Hello World'");
System.out.println("修改记录: Hello World");

//查询记录
sSql="SELECT cnt FROM test";
rs = st.executeQuery(sSql);
System.out.println("查询记录（中文内容）");
while(rs.next()) {
    System.out.println("\t"+rs.getString(1));
}

//删除记录
st.execute("delete from test");
System.out.println("删除记录: test");
```

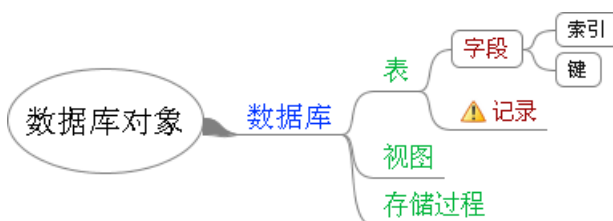
6.4 数据库模型

常用数据操作之外的其他操作，称为相关数据操作。数据库有如下对象：数据库、表、

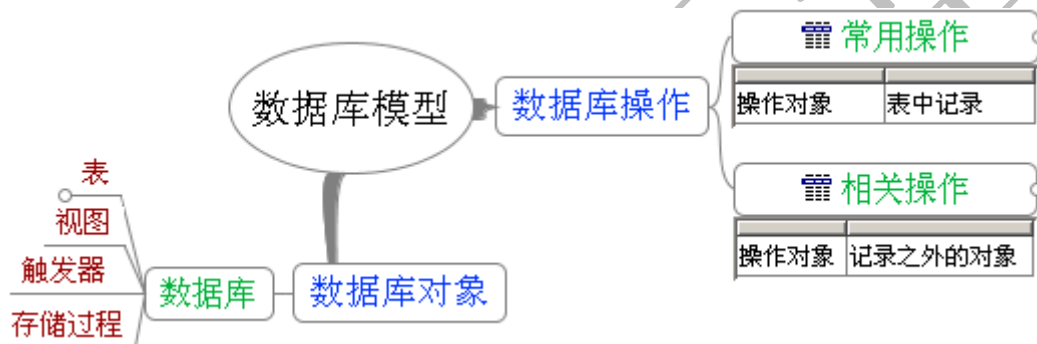
视图、存储过程、记录、字段、索引、键，除了记录之外，其他对象也需要增删改查，相关数据操作就是对这些对象进行处理。

在此不再深入讲述，只留下一个整体概念，数据库对象如图6-7所示，数据库模型如图6-8所示。

数据库原理，讲述了一堆又一堆的理论，一看就让人头痛。但对绝大多数人来说，简单概括如图6-8所示，数据库就分为数据库对象和数据库操作，而其他知识，除非要去进行这方面的研究，没有多大必要去浪费时间。



(图 6-7 数据库对象)



(图 6-8 数据库模型)

```

//创建表
sSql="create table test(cnt varchar(200))";
st.execute(sSql); //执行给定的 SQL 语句
System.out.println("创建表: test 成功!");

//删除表
st.execute("drop table test");
System.out.println("删除表: test");
  
```

6.5 释放资源

具体操作完成以后，必须释放资源，使其可以充分利用，而不致于耗尽资源。

```

//关闭数据库、数据库连接
st.executeUpdate("SHUTDOWN;"); //关闭数据库
st.close(); //关闭执行语句对象
conn.close(); //关闭数据库连接
System.out.println("关闭数据库、执行语句对象、数据库连接");
  
```

注意：一般操作不用关闭数据库，只有在应用系统关闭前，才需要关闭数据库。

```
st.executeUpdate("SHUTDOWN;"); //关闭数据库
```

6.6 小语句解决大难题，IT 需要简单化

一朋友碰到一个难题，请我帮他写个小程序。要求如下：

有一个 Excel 文档，三张表格：NBS（近 7 千条）、Sheet1、CUSTOM（2 万条）。里面内容是国内一些企业的资料，要求找出 NBS 和 CUSTOM 两张表格里同时出现的企业，即找出名称相同的企业，然后将企业名称和两个企业代码复制到 Sheet1 里面。

那朋友先是手工操作，找了 200 多条，估计花了许多时间。但因为实在太多，只得请求帮助。

我的做法如下：

选中工作表 Sheet1，从菜单中选择：工具 -> 宏 -> Visual Basic 编辑器
copy 以下代码，保存后关闭，任意双击 Sheet1 的一个单元格即可。

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
    Dim cnn, rst
    Set cnn = CreateObject("ADODB.connection")
    Set rst = CreateObject("ADODB.recordset")
    'nbs 列名: _COL0, _COL1
    'custom 列名: PARTY_ID, PNAME

    cnn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Extended Properties='Excel
8.0;';Data Source=" & ThisWorkbook.FullName

    Sql = "select b1.[_COL0],b2.PARTY_ID,b1.[_COL1] from [nbs$] as b1,[custom$] as
b2 where b1.[_COL1]=b2.PNAME"
    rst.Open Sql, cnn, 1, 1
    Sheets("Sheet1").Range("a1:c10000").ClearContents 'Sheet2 为测试用的工作
表，换成自己的
    Sheets("Sheet1").Range("a1").CopyFromRecordset rst
    rst.Close
    cnn.Close
    Set rst = Nothing
    Set cnn = Nothing
End Sub
```

也许很多人不懂 VBA，但这是最简单的解决方式。其实用什么语言来编程，不是重要的。语言只是思维的体现，你解决问题的一种方式而已。所以我们更需要的是培养自己的思维，培养自己解决问题的能力。

拿以上的 VBA 来说，除了这一句：Sql = "select b1.[_COL0],b2.PARTY_ID,b1.[_COL1] from [nbs\$] as b1,[custom\$] as b2 where b1.[_COL1]=b2.PNAME" 之外，其他都是固定的，你也不用去了解为啥要这样写。就如使用操作系统一样，你只要会用，而根本不需要明白操

作系统为啥能那样用。呵呵，你可以问问自己，你除了能使用之外，还了解多少操作系统的知识。道理一样，编程时，我们很多时候只要知道用什么能解决问题即可。

注：[_COL0]指标题为“_COL0”的那一列，如果没有标题，就用 F1 代表第一列，其他类推。

就拿我自己来说，很少用 VBA，还是 99 年才用过半年的 VB，但我知道用 VBA 能解决问题，于是去 Google 资料，绝大多数人用的是 vlookup。7000 条数据，如果用 vlookup，需要很长时间，所以我就查使用 SQL 的方式。

这种方式看起来代码不少，但很多代码你根本不用去管，照样 Copy 过来，改一下你要查什么内容，即修改一下 Sql 的定义（这 SQL 语句的意思很简单，select 你要的东西，from 你要获取数据的地方，where 你需要取的范围。不要觉着 SQL 语句高不可攀，其实这就和你想事情的方式一样），再改一下你要把数据放在哪里就行了。

IT 需要简单化，上面讲的方法很容易掌握，我相信你以后能用这解决更多问题。

后记，为了告诉朋友怎么解决问题，写了一点简短的说明（内容为做法部分），并把这点说明发表为《Excel 用 SQL 来查询，小语句解决大难题》（做法加需求），本节对此文进行深度挖掘，其实 IT 很简单，只是很多人把 IT 搞得复杂了而已。

第 7 章 人生没有救世主，一切靠自己

不要指望别人能救你，也不要奢求他人会给你带来明灯，更不要梦想上帝会眷顾你而让你不劳而获。谁是你的救世主？人生没有救世主！谁也不是你的救世主！如果真要有的话，只有你自己才是你的救世主，也只有你自己才能救得了你自己！

救自己的方式只有一个，那就是勇敢地战胜和超越你自己，必要时放下一切重头再来。

给你自己一个目标吧！那怕很小的一个，只要你努力，就能达成。在你上一个目标达成以后，再给自己定下一个目标，再努力去做！

当你有了勇气和自信，再去完成又一个目标。一步一个脚印往前走，别只看着那些已经完成的目标，为自己定一个不断前进的征程。别给自己留下懒惰的时间和空洞的幻想，不停地走，你将发现你已经救了自己！你也将慨叹你的勇敢和运气！不是吗？你已不再怯懦，不再畏缩，你已经有了一种豪气和挑战一切的勇气。

你成功了吗？没有！人生不会有所谓的最终的成功，有的只是你对人生的豁达和你看事物的那份坦然。人活的是一份良知和一份责任，活的是一种平凡和一种心态！

曾几何时，风险投资、VC 等词充斥耳边，创业者谁不希望天道酬勤，获得一笔投资，为自己的事业装上助推器？要想受到投资人的青睐，获得投资，商业计划书是第一步。

可商业计划书怎么写，具体有哪些内容？其中经营计划最为重要，一个酝酿中或没有明确经营计划的项目，往往很模糊。通过制定经营计划，使用 FreeMind 将整体思路写下来，将你的思维图形化，正反理由都写下来，然后逐条推敲，很快就能把经营计划做出来，经营者对该项目就有更清晰的认识。

企业是如此，商业计划书对个人的成长更有借鉴意义。个人成长可以分为多个阶段，每个阶段都存在以下 6 点：你现在想做什么？必须做什么！你以前做过什么？现在能做什么！以后要做什么？现在怎么做！如果能抓住以上 6 点，很多事情就能迎刃而解。

如我找对象那样，我看中的是对方孝顺。因为我知道，如果对方孝顺，起码本质不差。选择朋友及合作伙伴，更要注重对方的人品。

7.1 商业计划书的启发

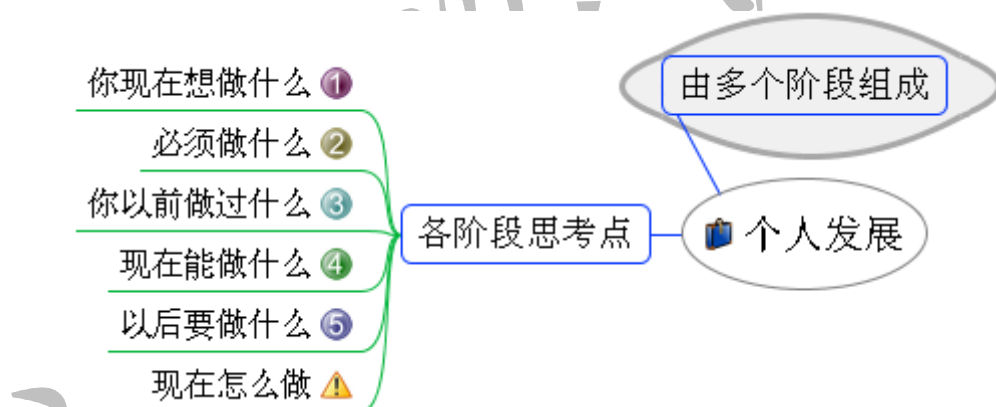
有段时间和朋友在操作一个项目，因为自己能写点东西，所以我们做了一个分工，他负责跑资金的事，我负责写商业计划书，所以我找了不少商业计划书的资料。注：这个项目后因投资太大而搁浅。

其中有一份标准的商业计划书范例，仅提纲就有 18 页，如果真按那样来写，将会是一个漫长的过程，可商业计划书真有这么复杂吗？

于是我继续找，找到了另一篇：商业计划书待解答问题（具体内容请参考：7.2 商业计划书待解答问题），一下就豁然开朗了。我把这些资料发给那朋友，再和他交流时，他说先由他来写这商业计划书（我正忙着写书）。因为有了这些问题点后，商业计划书的轮廓就出来了，所以他也能写了。

可见我们不管做什么事，要抓注问题的要点，纲举目张，自然很快就能把问题找出来，并且很快就能解决。如“7.2 商业计划书待解答问题”所示，商业计划书分为 15 个大点，对于个人的发展（简称为人生）来说，是不是也可以包括以下几点（由于自身时间及知识的局限性，在此不提个人的发展可以分为几点，只提可以包含几点）：

你现在想做什么？必须做什么！你以前做过什么？现在能做什么！以后要做什么？现在怎么做！



(图 7-1 个人发展)

从我多年的人生经历来说，如果能抓住以上 6 点，很多事情就能迎刃而解，特别是必须知道现在要做什么，如图 7-1 所示。本节内容有保留，只提供简单描述，详情 howwe.net 或咨询浩为公司。

对于学习一门技术，比如数据库开发，如果抓不住要点，自然会无从下手，即使勉强入门，也需要花费许多时间。本书的撰写，就在于抓住数据库开发的要点，期望能使其他人少走弯路，从而迅速入门。

7.2 商业计划书待解答问题

1) 你的眼光是什么？

* 你的远期目标是什么？

- * 你要解决什么问题？对象是谁？
- * 你将来想要成为什么样的人？
- 2) 你的市场机会是什么？市场有多大？
 - * 您目标的市场有多大？发展有多快？
 - * 这个市场有多成熟，或多不成熟？
 - * 你是否有资本成为这个市场前三位？
- 3) 介绍你的产品和服务
 - * 你的产品或服务是什么？
 - * 解决了用户的什么问题？
 - * 你的产品或服务有什么特别之处？
- 4) 你的用户是谁？
 - * 谁是现在的用户？
 - * 谁是目标的用户？
 - * 理想的用户是什么样的？
 - * 谁会付费？
 - * 介绍一下某个具体用户的例子
- 5) 你的价值主张是什么？
 - * 你给用户提供了什么价值？
 - * 使用 / 买你的产品，用户的投资回收率是什么？
 - * 你解决了什么问题？
 - * 你是销售维他命，阿司匹林，还是消炎药？（奢侈品，有益的东西，还是必需品？）
- 6) 你如何销售？
 - * 销售程序是什么？周期有多长？
 - * 你的销售和市场方针是什么？
 - * 你当前的销售链是什么？
- 7) 你怎么吸引客户？
 - * 争取每个用户要花费多少钱？
 - * 在不同时期这个费用是否不同？为什么？
 - * 用户的永久价值什么？
- 8) 你的管理团队有谁？
 - * 你的管理团队有谁？
 - * 他们有什么经验？
 - * 欠缺那些环节？有什么计划去弥补？
- 9) 你的收入模式是什么？
 - * 如何赚钱
 - * 你的收入模式
 - * 需要怎样才能盈利？
- 10) 你现在进展到哪一步？
 - * 你现在进展到哪一步了？技术 / 产品？团队？财务 / 营收？
 - * 现在进展情况如何？现状和前景是否更清晰了？
 - * 你将来的计划是什么？
- 11) 你的融资计划是什么？
 - * 已经得到了什么投资？
 - * 希望得到多少投资？比例如何？

- * 资金用在什么地方？
- * 资金可以支持多久？到那时公司是否可以发展到一个重要里程碑？
- * 你还打算吸引多少资金？什么时候？
- 12) 你的竞争对手是谁？
 - * 谁是你当前和潜在的竞争对手？
 - * 谁有可能和你竞争，谁有可能和你合作？
 - * 你的优势和弱点？
 - * 你有什么特殊之处？
- 13) 你有什么合作伙伴？
 - * 谁是你的销售或技术合作伙伴？当前？未来？
 - * 这些合作伙伴有多可靠？
- 14) 为什么适合有意的投资者？
 - * 和投资者的方向，经验吻合？
 - * 与投资者现有的投资组合有什么互补，或竞争？
- 15) 其它
 - * 成功的条件里有什么还只是假设？
 - * 有什么突然因素有可能一夜之间改变你的生意？新科技，新市场成员，规则法规的变化？
 - * 你公司的薄弱环节是什么？

7.3 自我经历：知道自己要做什么

从 2009 年 4 月底开始，在 CSDN 学生大本营做老师（义务劳动），有学生希望我能讲讲我的经历，于是有了这些内容，希望也能给你带来帮助。

一、小学及学前

由于人数不够，村上（当时叫大队）那年的幼儿班不开班，于是直接进了小学。以致我和别人开玩笑说，我幼儿园都没毕业。

我小时候很顽皮，上学后一段时间经常在课堂上吹口哨。更搞笑的是，我当时不会写字，作业都是求别人写的，尽管上学前家人也教过，可就是没学会。也不知多久后，脑子突然开窍了，就能写了。

小学给我的印象：我除了家庭作业按时完成之外，其他时间就是玩。不幸的是，小学毕业前还迷上了小说，别人在忙着准备升学考试的时候，我却看了杨家将、三国演义、水浒传等小说。也因为这个原因，东山中学（老毛曾在那读过，老毛是我对开国领袖毛主席的尊称，在我记忆中，很多湖南人都这么尊称）就差几分而没考上。

现在回想起来，如果当时有人能给我一些指导，也许我就不会在别人忙着准备考试，而自己却在看小说。

二、初中

初中读书很用心，成绩挺不错，初三曾拿过学区统考第三（三乡一镇十多个班）。但在初三住校期间，由于学校扩建，租住在农民家里，住宿条件很差，特别是湿气很重。因湿气引起手上、腰上长包，当时也没重视，再引起便血，从而贫血。反正中考后身体很虚，严重贫血，记得有个参数只有几克。

当时生病的一个好处是，我没考上中专，但也没考上东山（又差几分），给我的人生增添了许多磨难。如果我考上中专，估计就呆在老家了，最多也就混一个市或乡干部当当。

初中给我的印象：如果初三多和父母沟通或父母多关心的话，也许我不会生病。也就是说在成长过程中，人生需要引导。不过我从没在这事上怨过父母，因为我父母为我们操劳太多，我有三个哥两个姐，我爸妈的辛苦可想而知。

三、高中

由于生病，没考上东山中学，也没去差一些的四中，所以在八中浪费了将近 3 年时间。上课要么睡觉、要么和老师对着干，记得高一还在前三。93 年会考一结束就离校，去上海打工，但离校时我知道我还会读书。94 年春节后去补习学校读书，750 的高考总考了 480 多，于是很自信，直接在家复习。记得 95 年 3 月才去补习学校，以致高考时发挥严重失常。于是只得老老实实在补习学校呆了一年，尽管没有正常发挥，96 年还是考上了上海的东华大学。之所以说高中三年是浪费，因为 94-96 这三年时间（呆在学校 2 年不到）是我高中三年的一个轮回。

高中给我的印象：我很不懂事，也没有人能指导我该怎么做，但我坚强的走出来了。

四、大学

由于 N 年的磨难，在大学，我很明白我要做什么。在学习上，99 年 5 月前课余就泡在图书馆；在生活中，结交一些觉得值得自己交的同学，而和其他只泛泛而交；高层交往中，常去陪一老人聊天，尽管这老人没给我带来看得见的帮助，但通过和他的交流，我学到了不少为人处事的经历；可惜的是另一个老师在我毕业前突然去世了，要不然，这老师的人脉将会给我很大帮助。

99 年 5 月至年底，几乎天天呆在学校的研究所里，给老师做项目。当时用的是 VB，在项目中我先写一段代码作为范例，另一同学就照着做其他的。尽管这同学因受人指使，在我的第一家公司曾扮演了不光彩的角色，而断绝关系，不过还是很感谢他在研究所期间的协作及帮忙。

大学给我的印象：我知道我要做什么，也在全力做，绝不浪费自己的时间。

五、唯一的打工公司

2000 年元旦后，开始在一家台资企业实习，毕业后就去了这家公司上班，到 2003 年 2 月才离开，所以这是我打工生涯中的第一家公司，应该也是最后一家公司。

这家公司事情不多，工作很轻松，如果不是因为自己想有点事，这个地方绝对能让我过得很舒服，因为我 02 年年初就在张江买了一个三室一厅的房子。

工作之余，自己系统学习了不少知识。

这家公司给我的印象：积极主动、用心把事做好；和同事把关系处好，千万别以为自己年轻气盛，就不知天高地厚而骄傲自大，多向别人虚心学习，争取多积累一些资源。

六、自己的第一家公司：大汉三通

这家公司是和大学同学一起开的，2003 年 3 月成立，2006 年底退出。在成立之前，就有不少同学劝我，希望我不要和那同学合伙，说我会吃亏。不是我不了解这个人，而是我知

道，如果我真想做点事，必须先练练手。对于准备以后开公司的人，我多讲几句。

俗话说：亲兄弟明算帐，对于合作伙伴，事先必须把事情说明白，白纸黑字写得很清楚，口头的都是没用的、骗人的。比如我们在成立公司前，说好公司股份分 30% 的技术股和 70% 的资金股。公司成立后，提到曾说好的技术股，竟可以说没这回事。呵呵，其实这股权对我来说，也就是那么回事，无异于废纸一张。

选合作伙伴，必须注意对方的人品，就如我找对象那样，我看中的是对方孝顺。因为我知道，如果对方孝顺，起码本质不差。选合作伙伴，更是如此。

但这次练手的代价很大，03 年卖掉房子赚的 20 万及近 4 年的时间就这样没了，加上一个女朋友也没了，另外主动打了一场官司，如图 7-2 所示。不过对于要开公司的人来说，打一场官司很有必要，会让你在以后的经营中注意很多事情、避免很多不必要的损失。就一万多块钱的事，知道实情的人都觉得挺搞笑。这点钱也没啥，但打个官司却可以长很多经历，在开庭之余，我曾和对方的代理律师（大家都认识）说，“打个官司练练手，顺便长点经历”。具体参考《喻世明言新篇-人生需要引导》中的“4.12 案例分析”，浦东法院宣判后，反诉到上海第二中级人民法院，真好玩。



(图 7-2 股权转让纠纷)

给我的印象：人要成点事，大多会有所损失，要不然就成不了大事。但不管做什么，不管在什么情况下，都要静得下心来，用心去做事。

另一点是这个社会没有傻瓜，只有把别人当傻瓜的傻瓜。

七、自己的公司：浩为

2007 初，从上一家公司退出的股份在广西又成立了一家公司，待这家公司初步走上正轨后，4 月我回上海，成立浩为。经过 2 年的经营，浩为目前开始步入快速发展阶段。

和合作伙伴见面时，总会阐述一番“整合优势资源，携手共进，共同发展”的观点，很受伙伴们欢迎。

我的总结是：用心做事，天道酬勤。

总之：人一定得知道自己要做什么。在人生的每一个阶段，争取让自己达到最佳境界。

7.4 谋事在人、成事在天，学会资源整合

看过许多与曾国藩相关的书，给我印象最深的一句是“纵有十分功劳，也要让天八分”，Google 查了很多次都没查到来源，结合曾国藩的“人天各占一半”，应该是“纵有十分功劳，也要让天五分”。在一次查询中发现《曾国藩从政为官方略》很不错，以下是内容提要：

第一讲 把脉时代“变局”，营造发展舞台

1. 面对“变局”，不作壁上观
2. 巧借“亦官亦绅”的双重身份
3. 脱胎换骨，打造湘军

第二讲 执理不移，遇事敢争

1. 卧薪尝胆，不轻易出底牌
2. 刚柔互用，争让适度
3. 舞台与作为，无权断不能有为

第三讲 大业凭众人智慧而完成(上)

1. 鉴人有术，冷眼识英才
2. “合众人之私以成一人之公”
3. 场面做大，才能群雄影从

第四讲 大业凭众人智慧而完成(下)

1. 广纳群言，倾听幕僚的意见
2. 认真对待反对者的声音
3. 传承事业，嘱意光大门庭之人

第五讲 曾国藩事上司之法

1. 倚人而起，不着痕迹
2. 功归于上，己能安享
3. 危难时也不树敌，巧为周旋

第六讲 英雄援手，大事易成

1. “引用一班正人”
2. 做大事要明剖是非，不可暗斗
3. 英雄要屈身

第七讲 立人达人之道

1. 自立立人，自达达人
2. 试以艰危，责以实效
3. 关心和尊重属下
4. 待属员不可太谦
5. 同患难，更当同富贵

第八讲 高官为政，务持大体——曾国藩做高官之道

1. 整顿官场积习
2. 首要之地，为官当守六大原则
3. 京信常通，三种情况决不上报

第九讲 成大事者以人才为本——曾国藩选拔人才的艺术

1. 选人不可以眼光太高
2. 成大事者需要有血性之人
3. 德才兼备，以德为本

第十讲 曾国藩的用人艺术之一：用人者应该具备的三大素养——“诚意、智慧、度量”

1. 要有用人的诚意
2. 要有知人的智慧
3. 要有容人的度量

第十一讲 用人者应该把握好的三大手段

1. 注重与人才的情感交流
2. 离不开利益的引导
3. 用人之道要宽严结合

第十二讲 成功用人的关键环节——充分发挥人才的特长

1. 考察是用人的基础
2. 将人才放在正确的位置上
3. 任人惟亲是用人的大忌

第十三讲 不可泄了沉毅之气——曾国藩应对挫折的三个方略

1. 把挫折看成是磨砺自己个性的机会
2. 要知道自己败在哪里
3. 要耐心等待形势的变化

第十四讲 曾国藩的为政艺术之一：保持一颗清醒的头脑

1. 居官者有四大败身之道
2. 名利二字要看得淡一些
3. 情绪化的反应是一种不成熟的表现

第十五讲 曾国藩的为政艺术之二：领导者贵在有战略头脑

1. 有战略头脑是高明领导应有的素质
2. 要有将战略决策执行到底的决心

第十六讲 处理好与家人的关系——曾国藩的六条齐家格言（上）

1. 不要有代代做官的想法
2. 不要沾染上官宦人家的习气
3. 居家之道，不可多有余财

第十七讲 处理好与家人的关系——曾国藩的六条齐家格言（下）

4. 盛时常作衰时想，上场当念下场时
5. 宦宦人家，不要干涉地方的公事
6. 仗势欺人是败家之道

第十八讲 良好的品性是成功的前提——曾国藩的修身艺术

1. 男儿自立，必须有倔强之气
2. 少年当有狂者进取之趣
3. 从砥砺自己的品性开始。

尽管此书题为《曾国藩从政为官方略》，但也能给个人的成长带来启发及指点。我们要

培养自己像曾国藩如何借助时局，营造自我发展舞台；如何集中众人智慧成就大事的本领。

曾国藩是一个十分务实的人，他深知自己讲理学、佐朝政尚可称职，而对于兵法阵战、练兵打仗却是个外行，但他充分利用资源，成就了一番伟业。以致毛泽东“独服曾文正”，蒋介石也对曾国藩推崇备至。

曾国藩最初出山办团练，以丁忧侍郎的身份“入局”。丁忧是为他母亲守丧，按儒家的礼制，无论官做多大，必须尽孝道，为父母守丧 3 年（实际是 27 个月）。刚开始在长沙“非官非绅，办事不灵”，到衡阳后“亦官亦绅，办事必成”，曾国藩利用他的名声、出身及人品，广招英才贤士，并通过各种方式结交地方乡绅，终于练成一支“忠诚”、“血性”之师。

曾国藩四大弟子之一薛福成的评价“是文正以团练始，不以团练终，且幸其改图之速，所以能成殄寇之奇功，扩勇营之规制也。”他还说曾国藩虽以团练发轫，“然惟早变其实，并变其名，所以能有成功，否则前事可睹矣。”

由于曾国藩打造了湘军这条船，而且湘军出征不久就把武汉打了下来，他的突出表现让倚食国家饷禄的正规部队汗颜。但湘军这条船实在造得太大、太显眼，磨难也就接连不断。直到 40 多岁时，他还说自己是“命薄福浅之人”，不足以当大任。尤其是每当关键时刻，总有大的突发事情降临到他的身上，为此他说“事机不顺，似天在磨折”，这或许是他后来总结的成大功大名“人天各占一半”的由来。

对于成长中出现的磨难，我们应该坦然处之，全力去克服。“故天将降大任于斯人也，必先苦其心志，劳其筋骨，饿其体肤，空乏其身，行拂乱其所为，所以动心忍行，增益其所不能。人恒过，然后能改。困于心，衡于虑，而后作。”孟子的这句话在两千多年前就阐明了这个道理，当今社会更值得我们去思索。

7.5 人生没有救世主，一切靠自己

不要指望别人能救你，也不要奢求他人会给你带来明灯，更不要梦想上帝会眷顾你而让你不劳而获。谁是你的救世主？人生没有救世主！谁也不是你的救世主！如果真要有的话，只有你自己才是你的救世主，也只有你自己才能救得了你自己！

救自己的方式只有一个，那就是勇敢地战胜和超越你自己，必要时放下一切重头再来。每个人都有自己的人生和思想，每个人都有自己的生活和生存方式，但每一次选择或抉择都是一次命运的考验与超越自我的尝试，不管是成功还是失败，没有人能来代替你自己，一切只能靠自己。

你的父母能在你年轻的时候帮你，但不能帮你一辈子，也不可能帮你一辈子。而且当你父母年老时，他们需要你的帮助，也只有你才能让你父母安度晚年。

有人说，父母年老，能靠政府养老，对于城市里有养老金的人暂时是可以这么说，但那些没有养老金的人呢，还有广大的农民兄弟呢？其实养老制度也许更是一个天大的谎言。

先看看美国：在 1950 年时，平均每 16 个工作人口支持 1 个退休人口，而 2004 年底，每 3.3 个在职者支持 1 个退休者，而且 20 年后，将是每 2.3 个在职者支持 1 个退休者。2030 年 65 岁以上人口所占比重将达到 20%。美国目前没有法定退休年龄，何时退休全看个人买的是哪种养老保险。享受社会保障福利的退休年龄从 65 岁延迟至 67 岁，2005 年美国国会参议院财政委员会建议将退休年龄延迟到 69 岁。据美国退休人协会表示，目前年龄超过 60

岁的退休人口约 4500 万，其中 46% 的老人，由于经济原因必须在退休后仍继续工作（包括半工在内）。即便如此，社保仍然面临危机。2005 年美国总统布什指出：“2018 年，社保将会出现入不敷出的情况。自那以后，社保缺口将逐年递增。到 2042 年，整个社保系统将被消耗殆尽，全面破产。”

美国是这样，中国呢？现在是 9 个劳动年龄人口（并不是真实劳动人口）对应一个 65 岁以上老年人口，40 年后是 2 个劳动年龄人口对应一个老年人口。现在只有城市的 4000 万老人享受社会养老金，并且是全国 13 亿人口通过不同途径贡献的（有人说要是没有数亿农民工的贡献，城市社保立刻崩溃），现在城市老年人的相对待遇是“前无古人，后无来者”，有些五十多岁就可以退休（工作二十几年，靠别人养二十几年），即便如此，现在的养老金缺口仍然相当于一年的国民经济总收入。

美国战略与国际研究中心（CSIS）2009 年 4 月 22 日发布报告称，到 2030 年时，中国将是比美国还要老龄化的国家。该报告指出，2005 年相对每 100 名适龄工作成年人，仅有 16 名中国老年人。但这一老年抚养比到 2025 年将达到 32%，到 2050 年将达到 61%。由于子女数量减少和城市化的进展，这些老年人都得依赖社保养老。

也许这些数据让你一时难以理解，就从你自己的情况来说吧。现在大多是独生子女，结婚以后，夫妻俩得供养两对父母 4 个人，其实他们的父母还有父母，这暂时不考虑。当你的子女成家后，他们要供养的人是 6 对父母 12 个人。

也就是说，每个人的负担只会越来越重，直至无法承受，那时就会有悲剧发生。

为啥要说这些呢，我只是希望能让大家明白，其实人的负担很重，也就是你说根本没有退路。要立足于社会，就必须提高自己的能力，一切得靠自己，当你有能力改变自己的时候，如果还不抓住机会改变自己，那以后等待你的是什么，可想而知。

给你自己一个目标吧！那怕很小的一个，只要你努力，就能达成。在你上一个目标达成以后，再给自己定下一个目标，再努力去做！

当你有了勇气和自信，再去完成又一个目标。一步一个脚印往前走，别只看着那些已经完成的目标，为自己定一个不断前进的征程。别给自己留下懒惰的时间和空洞的幻想，不停地走，你将发现你已经救了自己！你也将慨叹你的勇敢和运气！不是吗？你已经不再怯懦，不再畏缩，你已经有了一种豪气和挑战一切的勇气。

你成功了吗？没有！人生不会有所谓的最终的成功，有的只是你对人生的豁达和你看事物的那份坦然。人活的是一份良知和一份责任，活的是一种平凡和一种心态！

他人小故事：从来就没有救世主

国际歌中有一句唱得非常经典，“从来就没有救世主”，很多人早就听说过这句话，自己也会说这句话。但真正体会这句话份量的却是我过了而立之年。小时候，一直以为这个世界花香鸟语，是温暖的家园，所有的人都相亲相爱，直到生命结束。

在母亲的呵护下，我过着天使般的童年。不知道什么叫忧伤，也不知道什么叫勾心斗角，只知道这个世上人人都和我的家人一样，心地善良，到处是阳光。

中学时，好友萍离开我去外地上学，让我知道了什么是友情可贵。二十多年过去了，我们依然相处如初，尽管不在一起但从没间断联系，友情的花朵一直开在彼此心田，萍让我知道，友情没有救世主。

同窗六载相处十多年的朋友华章，在新世纪即将到来的一个没有星星的夜晚，伴着香甜

的梦，被一个比他坐的车还大的车吞到腹下，他没有任何知觉被送到了另一个世界。消息传来，我和同学们与天同泣。他走的这样匆忙，如果知道这样，真该时常问候他，让他走时不那么孤单。要知道，原本残疾的他得到的白眼儿比微笑要多很多。他的好日子才刚刚开始，年青的他，就这样永远告别了人世。他让我懂得世上没有永远，生命不是救世主。

当我满腔热情面对工作、爱情和生活时，一段苦涩的经历，让我开始懂得，生活不是一首动听音乐，回首茫然。忽然间，婚姻的外衣只剩我一个人穿时，大脑一片空白，仿佛夜归的人没了指路方向。婚姻的鞋，终点又回到了起点。我握着那张证明独身的纸，如同握着法码的秤杆儿，做事需用尺量，让远的近的，一切的目光让出距离，让背后少些言论的份量。在风雨袭来之时，我无法阻挡，只能用逃避来解决忧伤。

我曾无数次站在北京天安门广场，面对敬爱的毛主席像默默无言，他是人民的大救星，他明白我的心事吗？走在长安街头，我常常泪流满面，向前看，信心打折；向后看，打折信心。我不明白，人活着究竟为何？是为了遭罪吗？我小心翼翼维护着属于我的一切，难道上天给我的苦还不够吗？在无数个难眠之夜，我懂得了，生活没有救世主。

哥和姐相继下岗，不为别的，只因工作和领导有分歧，不和领导吃吃喝喝，在竞争上岗时，考试打了很高分，但领导用戴眼罩的手给打了考核最低分，单位人都知道，哥在单位里兢兢业业、正直无私、秉公办事。他是全单位唯一的会计师，如今遭遇下岗，让有能力的人才心寒。这样的领导不知在想什么？这样的改革有什么意义？这样的单位工作不做也罢，这种不器重人才的领导不侍候也好。塞翁失马焉知祸福？我相信有能力的哥，今后一定会开拓出一片属于自己的天地，我充满了祝愿。我想告诉姐姐，工作不是救世主。

姐去南方考察，我给看家。楼上楼下就我一个人，仿佛进了天罗地网的战场，随时都有敌人来袭。我像侦察员，把能开开的箱子柜子，包括桌子的抽屉都翻看一遍，直到确认无敌之后，又把大门小门全都关好，再把台灯拧亮，电话放枕头边，最后把菜刀拎在手上，插上卧室的门，钻进被窝，我自己一看也笑了，明亮的灯光下，一把明晃晃的菜刀和电话、枕头及一堆书籍放在一起，真是特别画面，大概对付小日本鬼子也不过如此。一看表，晚上九点多，该睡觉了，睡吧！越这样想着越睡不着，只好再看一篇小说，天啊，怎么会是破案还带点鬼的内容，我一下子恐慌起来。正在此时，只听“咕咚”一声，脑袋儿一下子大了起来，我感觉头皮发麻，手脚冰凉。我壮大胆子拎菜刀出去一看，原来是客厅我顶门的凳子上放的盆子掉下来，我一下子松了口气，真是自己吓唬自己。

于是，我又重新上床，但更没了睡意，就特别想有个人说说话。给谁打电话呢？在此时最需要关心鼓励的时刻，而心中只想也只是和一位多年在心底很有位置的挚友说说心里话。电话打过去，挚友的一句话让我心凉如雪，“我在和同学打麻将，不够手，有事吗？”我顿时无言以对，委屈的泪水止不住流下来。怎么这么傻，事过境迁，有几人会象我这样十几年不改变如此重情？当环境和地位发生变化时，人的心里不可避免地受到冲击和腐蚀，这是人的本性吧！我真的无言了，在这伤感的时刻，再没有比沉默更能表达我的心态了。从此，多年感情的翅膀折弯在麻将的快乐里。我不要再企求任何人，心化为冷漠，我懂得了，感情不是救世主。

此时，我反倒不再害怕了。一个人最大的恐慌莫过于生命的终结，如果一个人把生命都置之度外，还有什么可怕呢？就在这个惊慌的夜晚，我望着漆黑的夜，更深地体会到，无论生活、工作、爱情，都不是救世主。

经过几天的锻炼，从小不敢一个人独住的我，如今楼上楼下行动自如，没了惶恐，也没了心事。我知道自己今后一个人取暖的日子会很长，勇敢面对所有的苦难，也是人生一笔宝

贵财富，我坦然接受了。

我想告诉所有看到我这篇文章的人，世上从来没有救世主，自己才是真正的主人。

大洲点评：知道这一点易，明白这一点难，而身体力行难乎其难。生活中的许多困惑可能就源于这。人左右不了客观环境，但可以左右自己的主观世界。

浩为语：许多人在人生漫途中迷失了自我，被太多的不幸、苦难、坎坷所击垮，随波逐流，很少想到命运其实就在自己的手里握着呢。此文历数人生酸甜苦辣，从中悟出生命的真谛，因为来自于普通生活的感悟，相信会启发更多人，并如作者文末所祈盼的那样。

你成功了吗？没有！人生不会有所谓的最终的成功，有的只是你对人生的豁达和你看事物的那份坦然。人活的是一份良知和一份责任，活的是一种平凡和一种心态！

第 8 章 认识开源数据 HSQLDB

Hsqldb (HypersonicSQL)，一般简称为 hsql，是一款 Java 编写并开放源代码的免费数据库，支持标准 SQL 语法，完全支持 SQL-1992，可以自由使用和分发，目前已广泛使用。

相对其他数据库来说，使用简洁，体积小，运行仅需要一个 hsqldb.jar 文件，此文件不足 1M，就包括了数据库引擎、数据库驱动、用户操作界面等。

常用版本是 hsqldb_1_8_0，下载地址：<http://hsqldb.org>

中文文档下载：<http://code.google.com/p/hsqldb-zh-doc/>

运行 hsql 的模式有 Server 模式，进程内模式 (In-Process，也称之为 Standalone (独立) 模式) 和内存模式 (Memory-Only) 三种。

可以把 hsql 内置在应用程序中，使用户感觉不到数据库的存在，除了可以使用 In-Process (进程内模式) 外，最好是采用常用的 Server 模式，此模式下和 Oracle、DB2、Mysql、MsSQL 等数据库的操作方式一样。

8.1 hsql 运行模式

先了解一下 hsql 所涉及的文件，每个 hsql 数据库包含了 2 到 5 个命名相同但扩展名不同的文件，这些文件位于同一目录下。例如，名为“test”的数据库包含了以下几个文件：

test.properties、test.script、test.log、test.data、test.backup

properties 文件描述了数据库的基本配置。

script 文件记录了表和其它数据库对象的定义。

log 文件记录了数据库最近所做的更新。

data 文件包含了 cached (缓冲) 表的数据，

backup 文件是将 data 文件压缩备份，它包含了 data 文件上次的最终状态数据。

所有这些文件都是必不可少的，千万不可擅自删除。但如果你的数据库没有缓冲表 (cached table)，test.data 和 test.backup 文件不存在。

test数据库在运行时, log文件用来记录数据的变化, 在 hsql 正常关闭时被删除。如果非正常退出, log将用来在下次启动 hsql 时重复执行记录的更新操作。lck文件也可以用来记录数据库是否被打开, 它在 hsql 正常退出时被删除。有些情况下, hsql 会生成 data.old文件, 但随后会被删除掉。

注意: 当 hsql 引擎关闭数据库时, 它会产生扩展名为.new 的临时文件, 然后会把这些文件重命名为上面所列出的文件的名称。

此外, hsql 数据库可以链接到磁盘上任何地方任何格式化的文本文件, 比如CVS。

当SHUTDOWN 命令发出, 所有的活动事务(active transactions)都会被回滚(rolled back)。关闭数据库的一种特殊形式是通过SHUTDOWN COMPACT 命令。该命令重写.data 文件, 此文件包含了存储在CACHED 表中的信息, 该命令对该文件的大小进行精简(compact)。这个命令应该被周期性执行, 特别是在cached 表中执行了许多插入, 更新和删除操作时。数据库结构的变换, 比如删除、更改CACHE 表或者索引, 会产生大量未使用的文件空间, 可以通过此命令进行空间的回收利用。

一、Server 模式

Server模式提供了最大的可访问性(accessibility)。数据库引擎在JVM 里运行, 监听来自同一台计算机上或网络中其他计算机程序的连接请求。几个不同的程序可以连接到服务器并且查询和更新信息。应用程序(客户端)通过 hsql 的JDBC驱动连接服务器。在服务器模式中, 服务器最大可运行10个数据库。根据客户端和服务器之间通信协议的不同, Server模式可以分为以下三种:

1、Hsql Server

这种模式是首选的也是最快的。它采用 HSQLDB 专有通信协议。启动服务器需要编写批处理命令。Hsql 提供的所有工具都能以 java class 归档文件(也就是 jar)的标准方式运行。假如 hsqldb.jar 位于相对于当前路径的../lib 下面。命令如下:

```
java -cp ../lib/hsqldb.jar org.hsqldb.Server -database.0 mydb -dbname.0 demoDB
```

现在你可能会疑惑, [-database.0]、[-dbname.0]为什么在后面加[0]。因为服务模式运行时可以指定 10 个数据库, 如有多个数据库, 则继续写命令行参数-database.1 aa -dbname.1 aa -database.2 bb-database.2 bb。

新建文本文件保存上面命令, 文件名可以随意, 将后缀名改成 bat, 然后直接执行批处理文件即可。在以后介绍的执行启动工具的命令采用同样方法。也可以通过代码使 hsql 自启动和关闭。

上面启动服务器的命令启动了带有一个(默认为一个数据库)数据库的服务器, 这个数据库是一个名为"mydb.*"文件, 这些文件就是 mydb.Properties、mydb.script、mydb.log 等文件。其中 demoDB 是 mydb 的别名, 可在连接数据库时使用。

2、Hsql Web Server

这种模式只能用在通过 HTTP 协议访问数据库服务器主机, 采用这种模式唯一的原因是客户端或服务端的防火墙对数据库对网络连接强加了限制。其他情况下, 这种模式不推荐被使用。

运行 web 服务器的时候, 只要将刚才命令行中的主类(main class)替换成: org.hsqldb.WebServer

3、Hsql Servlet

这种模式和 Web Server 一样都采用 HTTP 协议, 当如 Tomcat 或 Resin 等 servlet 引擎 (或应用服务器) 提供数据库的访问时, 可以使用这种模式。但是 Servlet 模式不能脱离 servlet 引擎独立启动。为了提供数据库的连接, 必须将 HSQLDB.jar 中的 hsqlServlet 类放置在应用服务器的相应位置。

Web Server 和 Servlet 模式都只能在客户端通过 JDBC 驱动来访问。Servlet 模式只能启动一个单独的数据库。请注意做为应用程序服务器的数据库引擎通常不使用这种模式。

当 hsql 服务器运行时, 客户端程序就可以通过 hsqldb.jar 中带有的 HSQLDB JDBC Driver 连接数据库, 如采用 hsql 协议的默认端口 9002 时, 连接时不需要指定端口, 否则, 必须在 server 的 IP 地址或者主机名后面加端口。

java 代码

```
try{
    Class.forName("org.hsqldb.jdbcDriver") ;
}catch(ClassNotFoundException e){
    e.printStackTrace();
}
Connection c = DriverManager.getConnection("jdbc:hsqldb:hsqldb://localhost/xd",
"sa", "");
```

注: hsql 的默认用户是 sa 密码为空。修改默认密码的方法将在工具使用部分做出介绍。或直接修改 script 文件中 CREATE USER SA PASSWORD "" 部分, "" 中的内容, 注意密码为明码。

二、 In-Process 模式

In-Process 模式又称 Standalone 模式。这种模式下, 数据库引擎作为应用程序的一部分在同一个 JVM 中运行。对于一些应用程序来说, 这种模式因为数据不用转换和通过网络的传送而使得速度更快一些。其主要的缺点是, 默认不能从应用程序外连接到数据库。所以当应用程序正在运行的时候, 你不能使用类似于 Database Manager 的外部工具来查看数据库的内容。在 1.8.0 版本中, 你可以从同一个 JVM 的一个线程里面来运行一个服务器实例, 从而可以提供外部连接来访问你的 In-Process 数据库。

该方式的一个好处是可以不使用引擎, 在需要的时候操作数据。所以那些对数据库不是特别有要求, 但又需要操作数据库时, 使用这种方法。对于那些不想额外在数据库引擎花费金钱, 也可以使用这种方法。但是不推荐使用该方法。

推荐的使用 In-Process 模式方式是: 开发的时候为数据库使用一个 HSQLDB 服务器实例, 然后在部署的时候转换到 In-Process 内模式。

一个 In-Process 模式数据库是从 JDBC 语句开始启动的, 在连接 URL 中带有指定的数据库文件路径作为 JDBC 的一部分。例如, 假如数据库名称为 testdb, 它的数据库文件位于与确定的运行应用程序命令相同的目录下, 下面的代码可以用来连接数据库:

```
Connection c = DriverManager.getConnection("jdbc:hsqldb:file:testdb", "sa", "");
```

数据库文件的路径格式在 Linux 主机和 Windows 主机上都被指定采用前斜线("/"). 所

以相对路径或者是相对于相同分区下相同目录路径的表达式是一致的。使用相对路径的时候, 这些路径表示的是相对于用于启动 JVM 的 shell 命令的执行路径。

三、Memory-Only 数据库

Memory-Only 数据库不是持久化的而是全部在随机访问的内存中, 也叫内存数据库。因为没有任何信息写在磁盘上。这种模式通过 mem: 协议的方式来指定:

```
Connection c = DriverManager.getConnection("jdbc:hsqldb:mem:dbName", "sa", "");
```

你也可以在 server.properties 中指定相同的 URL 来运行一个 Memory-Only (仅处于内存中) 服务器实例。

注意事项: 当一个服务器实例启动或者建立一个 in-process 数据库连接的时候, 如果指定路径没有数据库, 那么就会创建一个新的空数据库。这个特点的副作用就是让那些新用户产生疑惑。在指定连接已存在的数据库路径的时候, 如果出现了什么错误的话, 就会建立一个指向新数据库的连接。为了解决这个问题, 你可以指定一个连接属性 ifexists=true 来限制只允许和已存在的数据库建立连接而避免创建新的数据库, 如果数据库不存在的话, getConnection() 方法将会抛出异常。

四、Server 模式和 web 程序如何搭配使用

可以采用两种方式:

1) 采用上面提到的方法, 运行 `java -cp hsqldb.jar org.hsqldb.Server -database.0 db/mydb -dbname.0 xdb` 来启动 db, 通过设置的 URL: `jdbc:hsqldb:hsqldb://localhost/xdb` 以达到在程序中访问 hsql 数据库。

(注: 设置 URL 等信息, 也可以通过 hibernate 来配置, 例如:

```
hibernate.dialect org.hibernate.dialect.HSQLDialect
hibernate.connection.driver_class org.hsqldb.jdbcDriver
hibernate.connection.username sa
hibernate.connection.password
hibernate.connection.url jdbc:hsqldb:hsqldb://localhost/xdb)
```

2) 为了省去第一种方法的琐碎, 可在启动 web 服务器时同时启动 hsql。具体做法如下: 编写一个有关 hsql 的 Listener 类, 扩展 `javax.servlet.ServletContextListener`, 可以在配置文件中设置 dbPath、dbName、port 等 hsqldb 的信息, 启动 hsql。详情请参考本章第 3 节。

在 hibernate 中, 使用 hsql 中, 如果设置主键类型用 `Idenity`, 那么 Java 的 field 用 `Integer` 类型, 不能用 `Long`。

当你启动数据库执行创建表, 增加数据的时候就会生成对应的 sql 语句。删除操作不会记录在 script 文件中。如果没有以正确的方式退出 hsqldb 那么会增加 *.lck 和 *.log 文件, 来记录你的操作, 直到你正确保存数据并且退出数据库, 这两个文件才会消失。

正确的退出数据库是执行 shutdown 命令, 如:

```
public class D10801ShutDown {
    public static void main(String[] args) {
        //定义变量
```

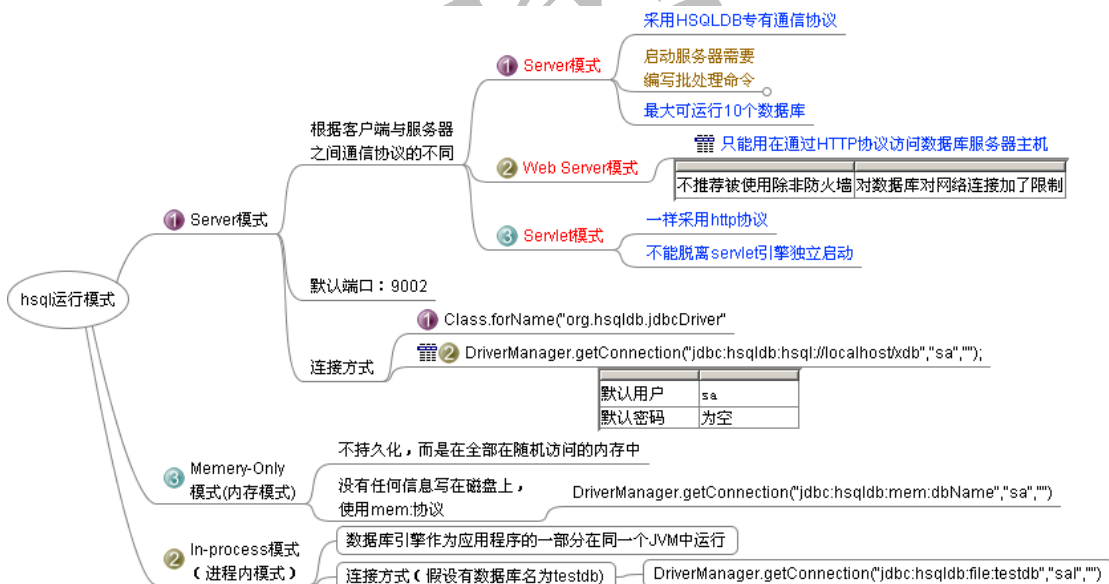
```

java.sql.Connection conn = null;    //数据库连接
java.sql.Statement st =null;        //执行语句对象
try {
    //创建数据库连接
    Class.forName("org.hsqldb.jdbcDriver");    //加载数据库驱动
    conn=java.sql.DriverManager.getConnection("jdbc:hsqldb:."
        , "sa", "");    //数据库连接，采用全内存访问（All-In-Memory）模式，
    即内存数据库

    st=conn.createStatement();    //创建执行语句对象

    //关闭数据库、数据库连接
    st.executeUpdate("SHUTDOWN;");    //关闭数据库
    st.close();    //关闭执行语句对象
    conn.close();    //关闭数据库连接
    System.out.println("关闭数据库、执行语句对象、数据库连接");
} catch (java.sql.SQLException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
}
}

```



(图 8-2 hsql 模型)

五、Hsqldb 脚本范例：

```

CREATE SCHEMA PUBLIC AUTHORIZATION DBA
CREATE MEMORY TABLE TSMSBOPERATOR(ID INTEGER GENERATED BY DEFAULT AS IDENTITY(START
WITH 1) NOT NULL PRIMARY KEY,NAME VARCHAR(30) NOT NULL,PASSWORD
VARCHAR(50),SERVICEIDLIST VARCHAR(500),MODULEIDLIST VARCHAR(500),PRODTYPEIDLIST

```

```
VARCHAR(500), DMPT VARCHAR(100), VERIFYTYPE TINYINT, IP VARCHAR(30), MTMAX  
INTEGER, MTNOW INTEGER, MTTIME VARCHAR(16), SMEMO VARCHAR(200), RESULT TINYINT, WTIME  
TIMESTAMP)  
CREATE UNIQUE INDEX PK_TSMSBOPERATOR ON TSMSBOPERATOR(NAME)  
ALTER TABLE TSMSBOPERATOR ALTER COLUMN ID RESTART WITH 12  
CREATE USER SA PASSWORD "SA1"  
GRANT DBA TO SA  
SET WRITE_DELAY 10  
SET SCHEMA PUBLIC  
INSERT INTO TSMSBOPERATOR  
VALUES(1, 'admin', '', '', '', 801, 802, 803, 901, 902, 903, 1001, 1002, 1003, 601, '', '', 1  
, '', 0, 0, '', 1, '2008-07-27 21:07:00.000000000')
```

以上脚本的简单说明（无说明的不用了解）：

- 1、创建表 TSMSBOPERATOR，类型为 MEMORY TABLE
- 2、在表上索引 TSMSBOPERATOR，类型为 UNIQUE INDEX
- 3、设置的主键的值为 12，ALTER COLUMN ID
- 4、设置用户“SA”的密码为 SA1，CREATE USER SA PASSWORD
- 5、添加一条记录

8.2 hsql 基本概念

hsql 从整体来讲，使用简单，但支持的数据有限，比如，我曾读取 10 万条记录插入另一个表，就没成功。还有不少功能使用很不方便，如 HSQLDB 没有使用索引来改善查询结果的分类，我查了很多资料，都没有具体介绍，勉强把索引加上了，却不能指定字段。总之，这个东西小数据还可以，而且特别方便，大数据就不行了，不过我没试过 TEXT 表，不知 TEXT 表性能如何。

1、表类型

HSQLDB 支持临时表和三种持久化表。

TEMP（临时）表是不写入磁盘，它仅仅只维持一个 Connection 对象的生命周期。每一个 TEMP 表仅对操作它的 Connection 来说是可见的，其他和数据库并发的连接只能访问自己的 TEMP 表备份。从 1.8.0 版本开始，TEMP 表的定义遵循 SQL 标准的 GLOBAL TEMPORARY 类型。表的定义是持久的，但每个新的连接只能看到它自己的表，而这些表刚开始是空的。当连接提交以后，临时表里的内容默认会被清空。如果临时表的定义语句里边包括 ON COMMIT PRESERVE ROWS，那么当连接提交发生时，临时表的内容将被保存。

三种类型的持久化表依次为 MEMORY（内存）表，CACHED（缓存）表和 TEXT（文本符）表。

Memory 表是使用 CREATE TABLE 命令的默认表类型。Memory 表数据全部驻留在内存中，但是对于表结构或内容的任何修改都被写入到<dbname>.script 文件中。script 文件在下次数据库打开的时候被 MEMORY 读取，里边的所有内容在 MEMORY 表中重新创建。所以跟 TEMP 表不同，MEMORY 表被默认为是持久的。

CACHED 表是在使用 CREATE CACHED TABLE 命令的时候生成的。它只有索引或部分数据

是驻留在内存中的，所以可以允许生成大容量表而不用占用几百兆的内存。CACHED 表的另外一个优点，即使它存储了大量的数据，数据库引擎只需花费很短的时间就可以启动。它的不足是在速度上有所降低。如果你的数据集相对小的时候，尽量不要使用 CACHED 表。在小容量和大容量表共存的实际应用中，最好对小容量的表使用默认的 MEMORY 表。

TEXT 表是在 1.7.0 版本中开始支持的，它使用 CSV（逗号分割数值）或其他分隔符的文本文件作为数据源。你可以指定一个已有的 CSV 文件（比如其它数据库或程序导出的数据）作为 TEXT 表的数据源，你也可以指定一个空文件用数据库引擎来填充数据。TEXT 表的内存利用效率比较高，因为它只缓存部分文本数据和所有的索引。TEXT 表的数据源如果需要的话，可以重新分配到不同的文件。TEXT 表详细内容请自己 Google。

在 Memory-Only 数据库里，MEMORY 表和 CACHED 表的声明都看视作为非持久化的内存表声明。这种模式中是不允许声明 TEXT 表。

2、约束和索引

HSQldb 支持 PRIMARY KEY, NOT NULL, UNIQUE, CHECK 和 FOREIGN KEY（依次译作主键、非空、唯一性、检查、外键）约束。此外，它还支持唯一性索引或普通索引。这些支持相当广泛，覆盖了多字段约束和索引，再加上外键的级联更新和删除。

HSQldb 在内部自动创建索引来支持主键约束、唯一性约束和外键约束：为每一个主键约束或唯一性约束创建一个唯一索引，为每个外键约束创建一个普通的索引。因为这些原因，你运行和使用 hsql 不应该在受这些约束的同一字段集内创建重复的用户自定义索引。这将会导致不必要的内存和速度开支。

索引对于提高查询速度是至关重要的，当使用连接到多个表的查询时，对于每一个表的每一个连接字段必须有一个索引。当使用范围或等式从句（比如：SELECT ... WHERE acol >10 AND bcol = 0）时，需要为在条件中使用的 acol 列创建一个索引。但是索引在 ORDER BY 从句或某些 LIKE 条件中是没有任何作用的。

根据经验，HSQldb 能够以超过每秒 100,000 行的能力来处理查询，如果需要的话，任何花费数秒以上的查询应该被检查，并且应该对表中相应的列添加索引。

HSQldb 没有使用索引来改善查询结果的分类（浩为语：即在查询时不能指定索引，如果指定了，就没法指定字段，我测试过），但索引在提高查询速度上起着至关重要的作用。如果一个表没有使用索引，进行 SELECT 查询时，表中所有的行记录都要被检查一遍。WHERE 从句中的列中，如果有一个列建立索引的话，那么查询操作就可能直接从第一个候选行记录开始，从而减少了要检查的行记录的数量。

索引在多个表之间进行连接操作的时候显得更重要。在 SELECT ... FROM t1 JOIN t2 ON t1.c1 = t2.c2 执行时，对 t1 中的行一个接一个的进行操作，来查找 t2 中没有与之匹配的行。如果 t2.c2 没有任何索引的话，那么，对于 t1 的每一行来说，要必须检查 t2 的所有行。然而有一个索引的话，在很短的时间内就能找到匹配的行。如果，查询(query)在 t1 上还有一个条件（例如，SELECT ... FROM t1 JOIN t2 ON t1.c1 = t2.c2 WHERE t1.c3 = 4），那么在 t1.c3 上创建一个索引的话，将不需要一个接一个的检查所有的行，并且将每返回一行的查询时间降低到 1 个毫秒以下。所以，如果 t1 和 t2 各有 10000 行记录，那么没有索引的查询将会进行 100,000,000 次行检查。如果在 t2.c2 上创建一个索引的话，查询次数将会降低到 10000 次行检查和索引的查找。在 t2.c2 上有另外一个索引的话，只需要检查 4 行就可以得到第一个符合条件的结果行。

3、多数据库配置

Server URL 格式如下：

jdbc:hsqldb:hsqldb://主机:端口/别名

其中主机为指定 server 的 IP 地址或主机名，要连接的数据库通过一个别名来指定。

这个别名是在 server.properties 文件中定义的一个小写的字符串，它指的是 server 文件系统上的一个实际数据库或者 server 上临时的内存数据库。

一下是 server.properties 或 webserver.properties 中的数据库样例。它定义了上面列出的数据库别名以及客户端的访问方法，其中涉及到不同文件和内存数据库。

database.0=file:/opt/db/accounts

dbname.0=an_alias

database.1=file:/opt/db/mydb

dbname.1=enrollments

database.2=mem:adatabase

dbname.2=quickdb

database.3=/home/dbmaster/lists

dbname.3=

注意：上面配置文件中 database.3 的/home/dbmaster/ 目录下的数据库文件 lists.* 与空别名相关联。

8.3 hsql 自启动

Web 应用中使用 hsql，Web 应用启动关闭时要同时启动和关闭 hsql server。使用 hsql 作为测试数据库，在执行单元测试的时候也要启动和关闭 hsql server。

Hsql 有几种启动方式，其中有 res,mem 等方式数据库结构都是只读的，hibernate hbm2ddl 会报 The database is in read only mode in statement [create]。所以还是使用独立的 server 模式，但是需要自己控制启动和关闭。

1、让 hsql 随 Web 应用启动和关闭

这个实际上就是一个 listener。代码如下：（位于目录 src 包 net.howwe.core）

为了缩减篇幅，代码略，请自己查看 howwe-ok 项目 net.howwe.core/HsqlListener.java，文件具体路径：D:\work\hocat\hocas-ok\src\net\howwe\core\HsqlListener.java。

在 web.xml 中加入以下内容：（指定数据库路径及配置 listener）

```
<context-param>
    <param-name>hsqldb.dbPath</param-name>
    <param-value>{webapp.root}WEB-INF/data/jdb</param-value>
</context-param>

<listener>
    <listener-class>net.howwe.core.HsqlListener</listener-class>
</listener>
```

数据库访问配置文件 db.properties 在源码目录 config 下，内容如下：

JDBC.Driver=org.hsqldb.jdbcDriver

```
JDBC.ConnectionURL=jdbc:hsqldb:hsqldb://127.0.0.1:19132/webapp
```

```
JDBC.Username=sa
```

```
JDBC.Password=sa1
```

```
JDBC.DefaultAutoCommit=true
```

2、数据库图形化访问

用惯了 mysql 的 mysql query browser，总是想着看数据库里面的内容，怎么办呢。你的系统启动起来以后，在 shell 里面执行如下命令：

```
"%JAVA_HOME%/bin/javaw" -classpath ../webapp/WEB-INF/lib/hsqldb-1.8.0.7.jar  
org.hsqldb.util.DatabaseManager
```

要注意把其中的 hsqldb-xxx.jar 的地址改成你自己的地址。启动之后按照你配置的数据库端口，名称等等去连接就可以了。

第 9 章 SQLExplorer 使用简介及 SQL 简介

SQLExplorer 是 Eclipse 的一个插件，用来管理数据库。SQLExplorer 插件提供了一个使用 SQL 语句访问数据库的图形用户接口 (GUI)。通过使用 SQLExplorer，能够显示表、表结构和表中数据，以及提取、添加、更新或删除表格数据。SQLExplorer 同样能够生成 SQL 脚本来创建和查询表格。所以，与命令行客户端相比，使用 SQLExplorer 是更优越的选择。

通过在 Eclipse 中加入一个 SQLExplorer 视窗，SQLExplorer 插件为 Eclipse 配置了一个访问数据库的 SQL 客户端。

SQLExplorer 当前版本为 3.5，有两个版本：Eclipse 插件、独立客户端 (Standalone Client)。

其中独立客户端是一 RCP 程序，可以直接使用，不用依赖于 Eclipse。

下载地址：<http://eclipsesql.sourceforge.net/>

数据库可简单归纳为一个模型，分为数据库对象和数据库操作，SQL 语句就是对数据库对象的操作语言。

9.1 安装 SQLExplorer

从下载地址下载 SQL Explorer Plugin，将文件解压后，分别将 features 和 plugins 目录中的所有文件拷贝到 Eclipse 中对应的 features 和 plugins 目录中，重启即可可用。

注意：features 一般不用拷贝，仅放 plugins 即可。

这是安装 Eclipse 插件的常用方法，当然也可以使用 link 方式安装插件，请 Google 相应文档。

注意：整合版本中已包含，不用安装。

9.2 配置 SQLExplorer

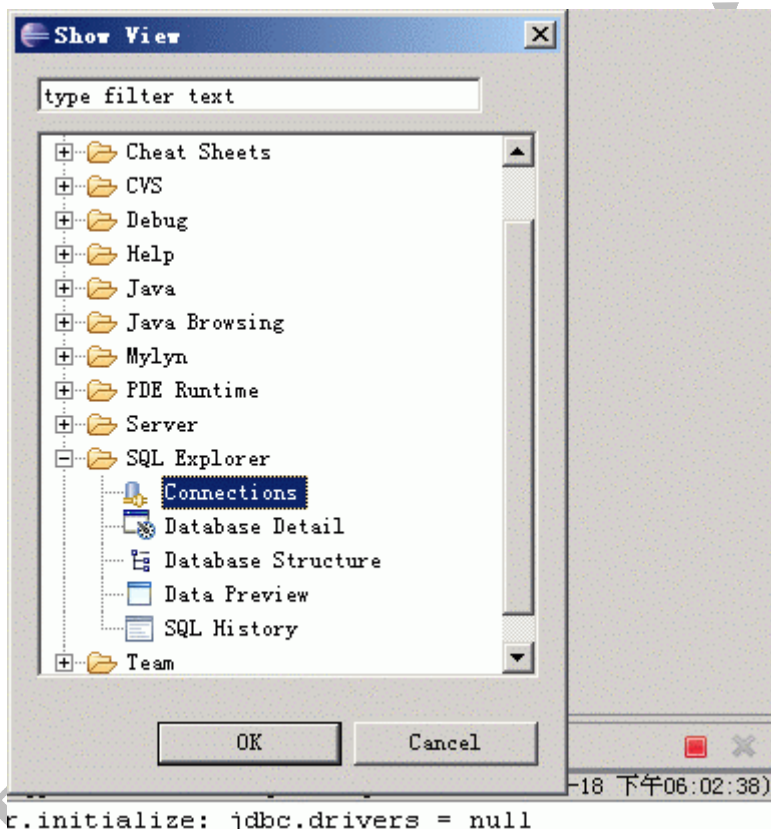
SQLExplorer 插件安装后，需配置后才能使用。下面，我们以 hsql 的配置为例。

1、运行 Web 服务器

运行 Web 服务器，主要是启动 hsql 数据库，便于以后的操作。

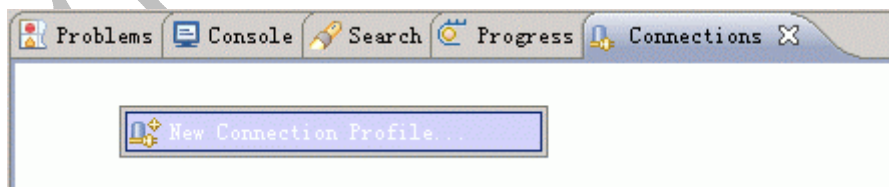
2、打开 SQLExplorer

在标签中打开 SQLExplorer，从菜单打开：Window => Show View => Other => SQL Explorer => Connections，如图 9-1 所示，选中 Connections 后选择 Ok。



(图 9-1 选择 Connections)

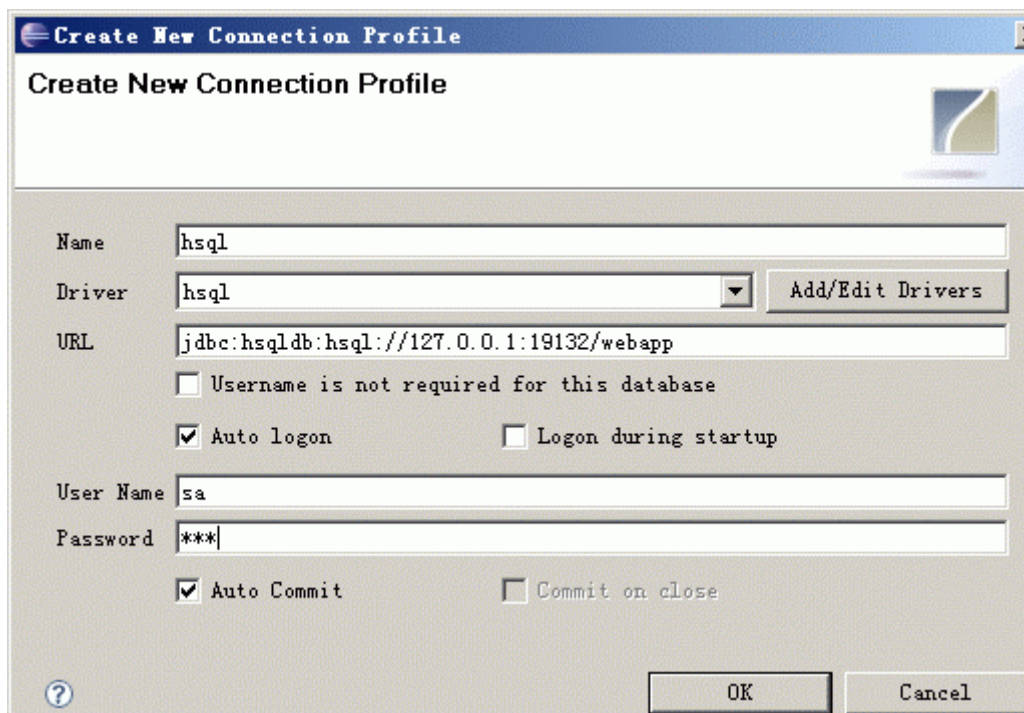
在 Eclipse 控制台区将出现一个标签 Connections，如图 9-2 所示，单击右键，选择 New Connection Profile。



(图 9-2 选择新建 Connection)

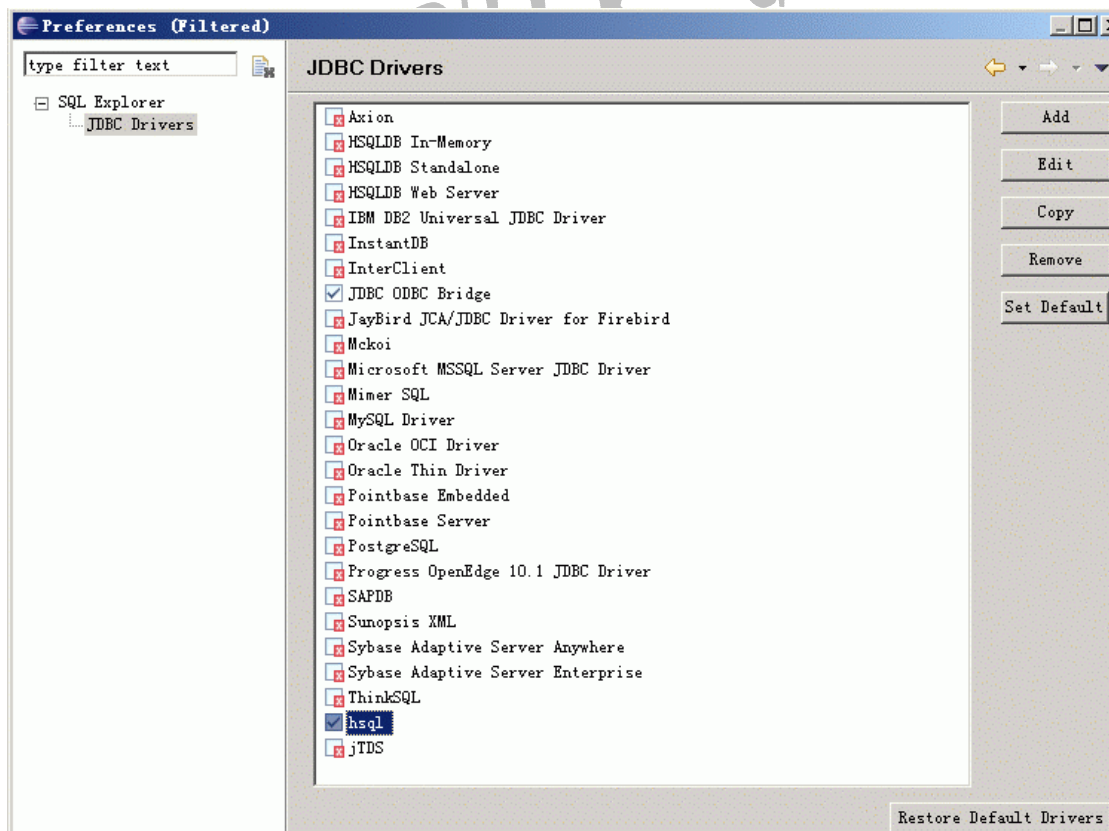
将出现如图 9-3 所示的界面，在 Name 输入 hsql，如果 Driver 下拉框已配置 hsql 驱动，则直接选择即可，否则需点击 Add/Edit Drivers 以配置 JDBC 驱动程序，如图 9-4 所示。

Driver 选择了 hsql 后，选中 Auto login，在 User Name 处输入 sa，Password 处输入 sa1，选择 Ok，将出现如图 9-6 所示的界面，这样配置完成。



(图 9-3 新建 Connection)

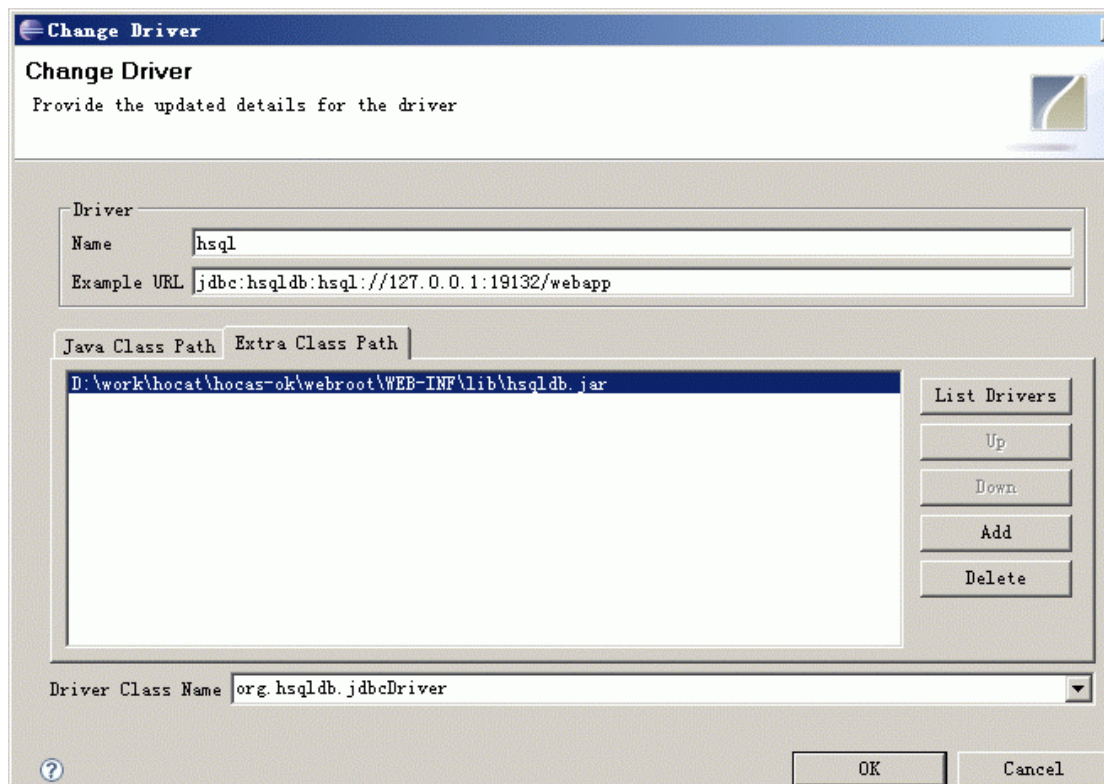
针对 hsqldb 数据库, 选择 hsql, 如图 9-4 所示, 双击或选择 Edit, 注意 hsql 由 HSQLDB Server 改名而来。



(图 9-4 选择驱动)

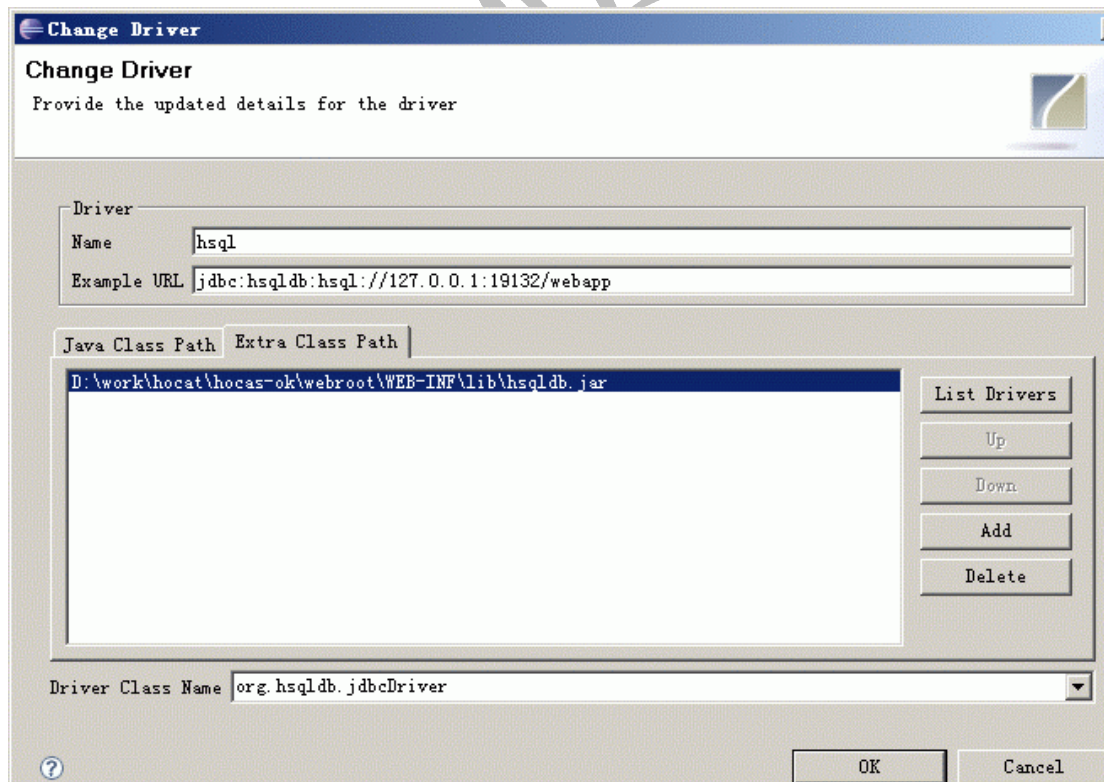
在如图 9-5 所示的窗口中, 选择 hsqldb.jar 的路径, 单击 List Drivers 后, 选择 Driver Class Name, 即选择驱动程序, 最后在 Example URL 处输入以下内容后, 确定即可:

jdbc:hsqldb:hsq://127.0.0.1:19132/webapp



(图 9-5 设置驱动)

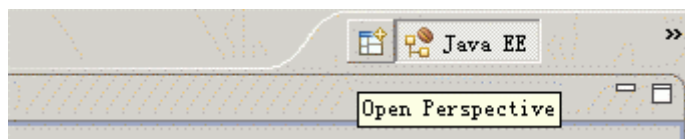
以上操作将一个 hsq 的连接加入到 connection 标签中，正如图 9-6 所示。想要修改一个别名，右击别名结点并选择 “Edit Connection Profile.”



(图 9-6 以添加连接及可进行的操作)

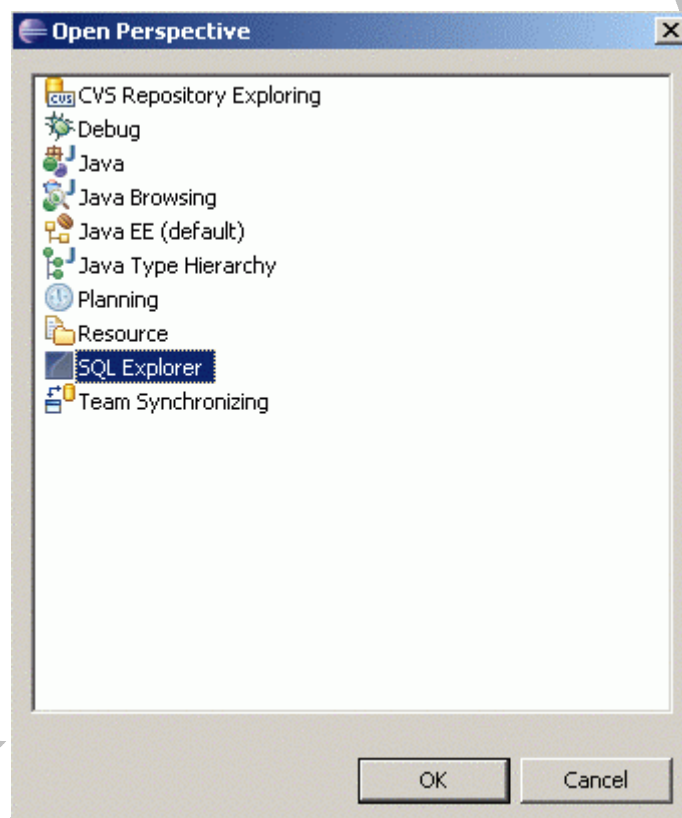
另一种方式是在视窗中打开 SQLExplorer，不过这种方式比较少用。

从菜单打开：Window => Open Perspective => Other 打开，或者在集成开发环境中单击“Open Perspective”按钮（如图 9-7 所示）来打开一个视窗。在条目列表中，选择“Other...”来显示 SQLExplorer 插件。



（图 9-7 Open Perspective 按钮）

在 Open Perspective 框中，选择 SQLExplorer 视窗，如图 9-8 所示。选择 SQLExplorer 视窗，再按上面所示的步骤建立连接，SQLExplorer 插件在 Eclipse 中就可以使用了。

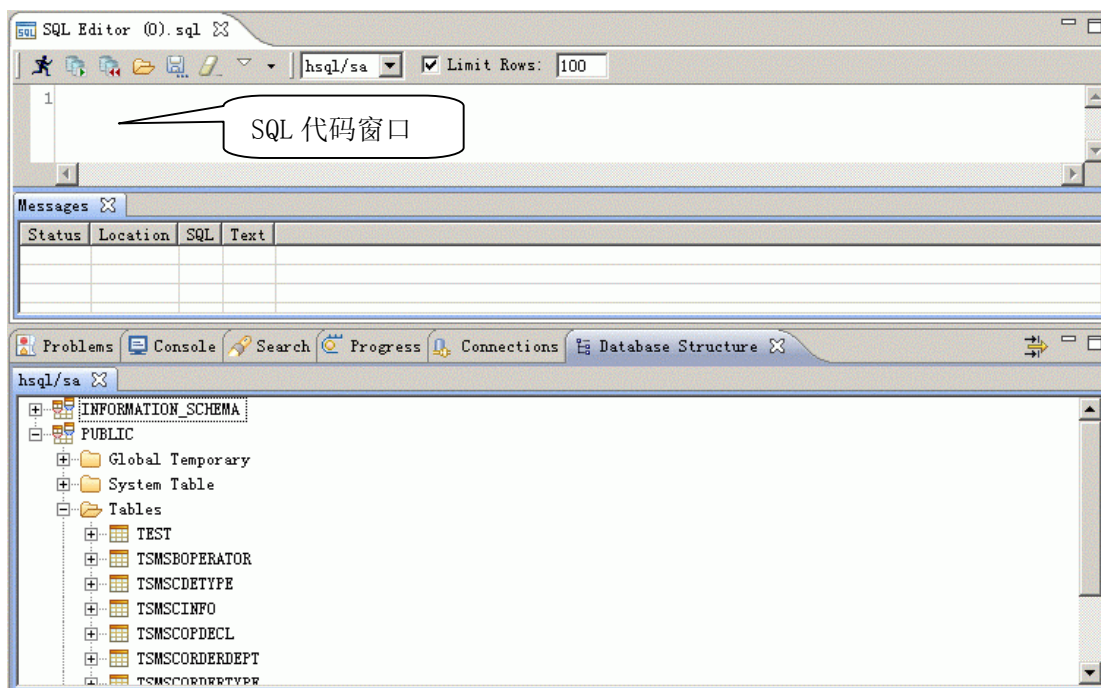


（图 9-8 Open Perspective 窗口）

Hsql 数据库的 JDBC 连接就这样建立好了。

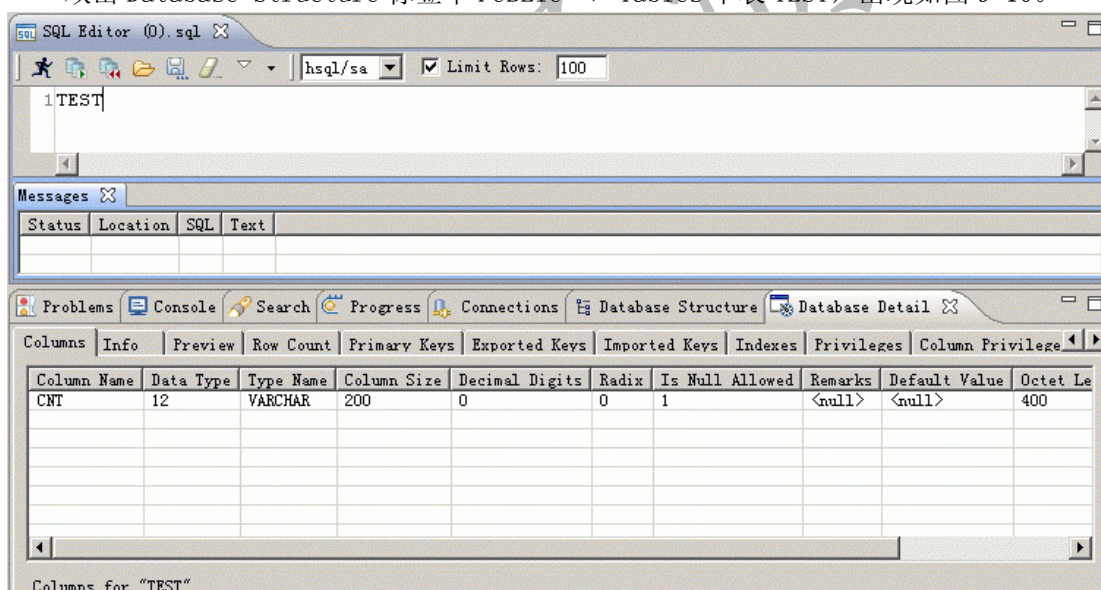
9.3 使用 SQLExplorer

在 connection 标签中双击 hsql 连接，或在 hsql 上单击右键后选择 New SQL Editor 将出现如图 9-9 所示的窗口，就可以进行 SQL 操作了。



(图 9-9 SQL Editor 窗口)

双击 Database Structure 标签中 PUBLIC => Tables 下表 TEST，出现如图 9-10。



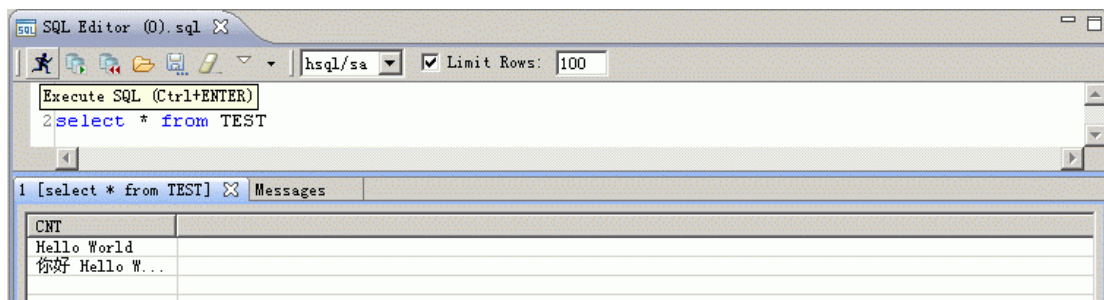
(图 9-10 DataBase Detail 窗口)

DataBase Detail 标签显示了表 TEST 的详细信息，同时在 SQL 代码窗口添加了表名 TEXT，以方便用户的操作。

1、查询

在 SQL 代码窗口输入如下内容：

```
select * from TEST
```

(图 9-11 查询)

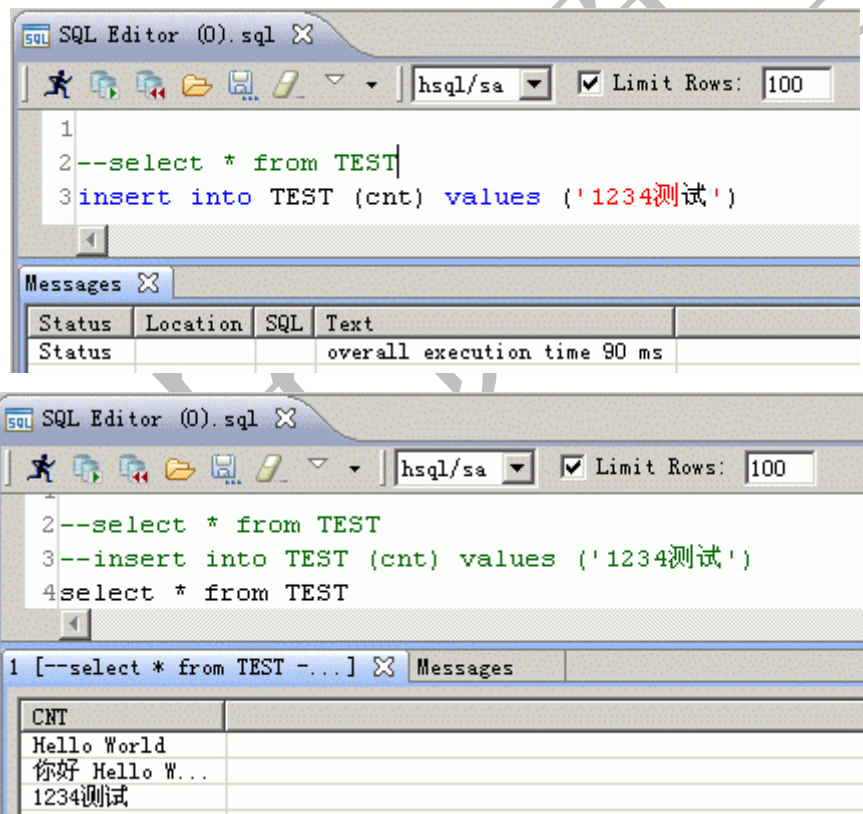
输入 sql 语句后，点击如图 9-11 所示的窗口中的人形按钮，执行结果如图 9-11 所示。

2、增加记录

在 SQL 代码窗口输入如下内容：

```
insert into TEST (cnt) values ('1234 测试')
```

注意：在代码前输入--表示将此语句注释，此语句不执行。



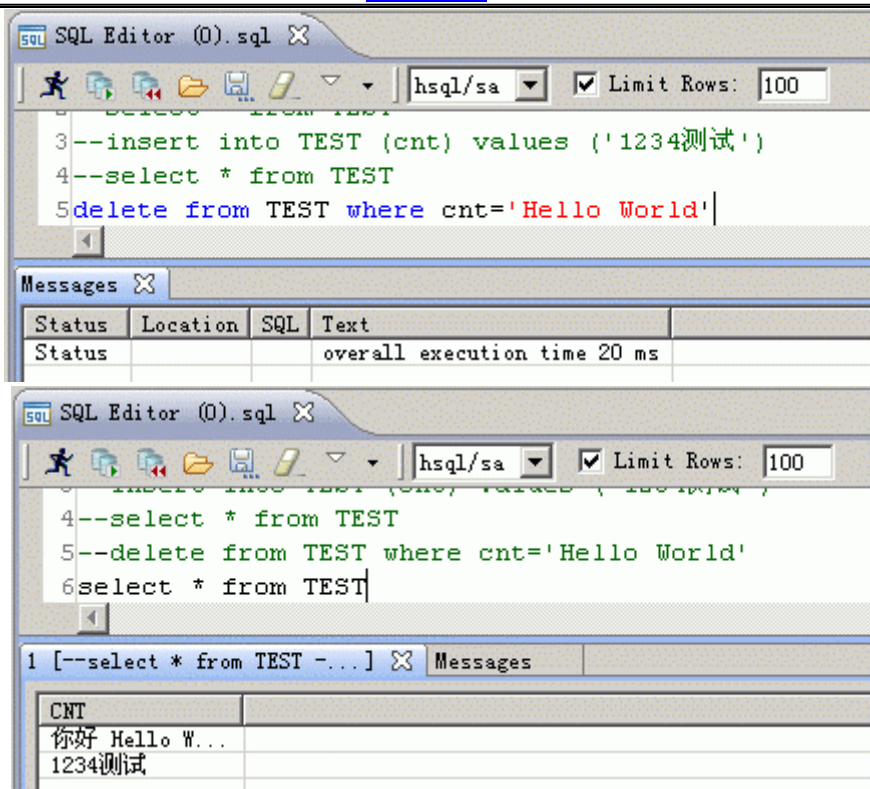
(图 9-12 添加及添加后的查询结果)

输入 sql 语句后，执行结果如图 9-12 所示。

3、删除记录

在 SQL 代码窗口输入如下内容：

```
delete from TEST where cnt='Hello World'
```



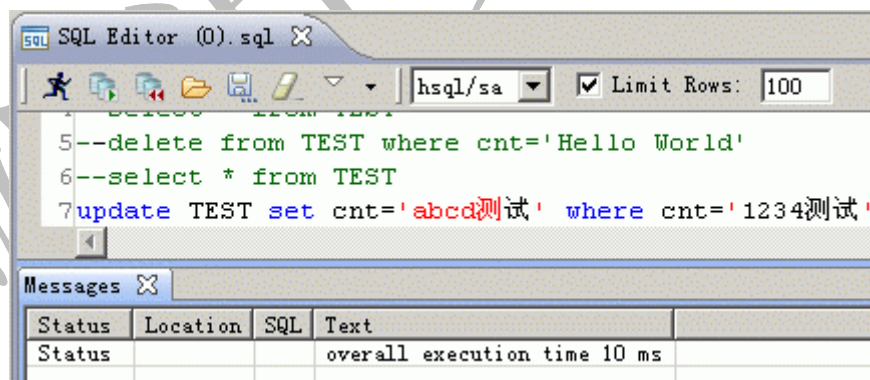
(图 9-13 删除及删除后的查询结果)

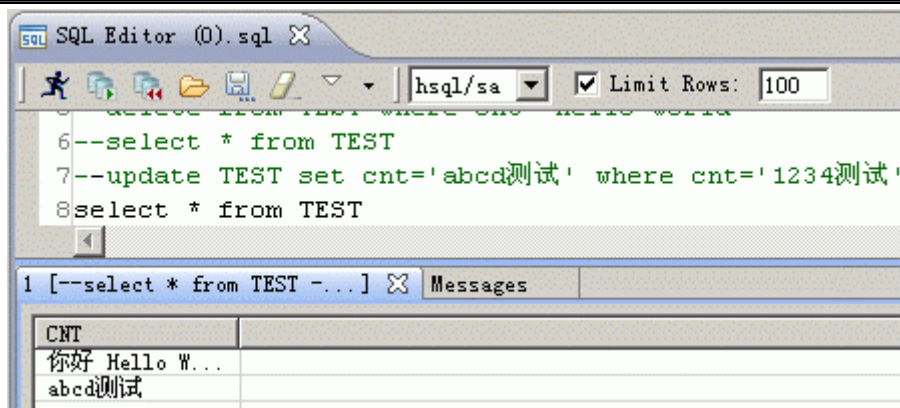
输入 sql 语句后，执行结果如图 9-13 所示。

4、修改记录

在 SQL 代码窗口输入如下内容：

```
update TEST set cnt='abcd 测试' where cnt='1234 测试'
```





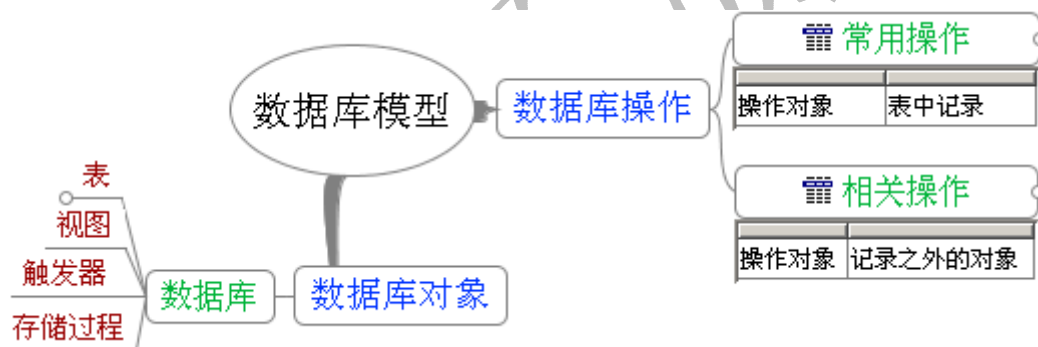
(图 9-14 修改及修改后的查询结果)

输入 sql 语句后，执行结果如图 9-14 所示。

9.4 SQL 基础

SQL 是用于访问和处理数据库的标准计算机语言。

正如前面所说，数据库就如图 9-15 所示的一个模型，分为数据库对象和数据库操作，SQL 语句就是对数据库对象的操作语言。



(图 9-15 数据库模型)

SQL 是一种 ANSI 的标准计算机语言，使我们有能力访问数据库的结构化查询语言。
(注：ANSI，美国国家标准化组织)

SQL 能做什么？简单一点就是能对数据库对象进行处理。

- 可从数据库查询出数据 select
- 可在数据库中插入新的纪录 insert
- 可更新数据库中的数据 update
- 可从数据库删除记录 delete
- 可操作数据库 create/alert/drop database
- 可在数据库中操作表 create/alert/drop table
- 可在数据库中操作索引 create/alert/drop index
- 可在数据库中操作存储过程 create/alert/drop procedure
- 可在数据库中操作视图 create/alert/drop view
- 可在数据库中操作触发器 create/alert/drop trigger
- 可以设置表、存储过程和视图的权限 grant/deny/revoke

其中常用的就四句：select、insert、update、delete，其余语句一般只有数据库管理员才用。

注：每种数据库都有自己的 SQL 数据类型及控制结构，具体请参考数据库帮助。

9.5 数据操作—针对记录

数据操纵语言 (DML, Data Manipulation Language)，四句话：select、insert、update、delete。

一、查询 select

简单的 SELECT 语句的语法如下：

```
SELECT [ALL|DISTINCT] <目标表达式>[, <目标表达式>]...
```

```
FROM <表或视图名>[, <表或视图名>]...
```

```
[WHERE <条件表达式>]
```

```
[GROUP BY <列名 1> [HAVING <条件表达式>]]
```

```
[ORDER BY <列名 2> [ASC | DESC] ]
```

含义：根据 WHERE 子句的条件表达式，从 FROM 子句指定的表或视图中找出满足条件的元素组，再按 SELECT 子句中的目标列表达式选出元素组中的属性值形成结果表。

如果有 GROUP 子句，则将结果按<列名 1>的值进行分组，该属性列值相等的元素组为一个组，每个组产生结果表中的一条记录。如果 GROUP 子句带有 HAVING 短语，则只有满足指定条件的组才予以输出。

如果有 ORDER 子句，则结果表还要按<列 2>的值升序或降序排序。

查询有连接查询、子查询（也叫嵌套查询）、合并查询（union）等几类。

如果一个查询需要对多个表进行操作，就称为连接查询。查询主要通过各个表之间共同的关联性来查询数据，它是关系数据库查询最主要的特征。

连接查询分为等值连接查询（连接条件使用=）、非等值连接查询（连接条件使用>、>=、<、<=、!=，及 BETWEEN...AND）、自连接查询（表使用 Join，同其自身进行连接）、外部连接查询和复合条件连接查询（WHERE 子句中使用多个连接条件，简称为复合查询）。

外部连接查询有两种：主表在左边，为左外部连接，简称左连接（Left Join）；主表在右边，为右外部连接，简称右连接（Right Join）。

注意：可以将 select 出来的结果生成一个新表，方法是在 from 前面加“into 新表名”。

二、增加 insert

插入单条记录：

```
INSERT INTO 表名称 VALUES (值 1, 值 2,....)
```

指定所要插入数据的列：

```
INSERT INTO table_name (列 1, 列 2,...) VALUES (值 1, 值 2,....)
```

增加多条记录：插入子查询结果或存储过程返回的结果。

三、修改 update

UPDATE 表名称 SET 列名称 = 新值 WHERE 列名称 = 某值

也可以修改多列数据：

UPDATE table_name SET 列 1=值 1, 列 2=值 2,... WHERE 列名称 = 某值

四、删除 delete

DELETE FROM 表名称 WHERE 列名称 = 值

9.6 数据定义-针对对象

数据定义语言 (DDL, Data Definition Language)，主要是对数据库对象的操作：数据库 (database)、表 (table)、索引 (index)、视图 (view)、存储过程 (procedure)、触发器 (trigger) 等，一般有创建 (create)、修改 (alter)、删除 (delete) 三种。

注意存储过程的执行一般用 EXECUTE。

详细情况请自己 google。

9.7 数据控制-授权

数据控制语言 (DCL, Data Control Language)，主要是管理用户能进行的具体操作，一般用户很少使用。具体情况请参考使用的数据库软件，本文不再展开。

9.8 MySQL 快速入门

如果想更详细了解 MySQL，请 google，本节仅简单说明 MySQL 的基本使用，让你快速上手。

一、下载

从以下地址可以下载绿色版 MySQL (此版本安装十分方便，很适合初学者)：

<http://howwe.net/tools>

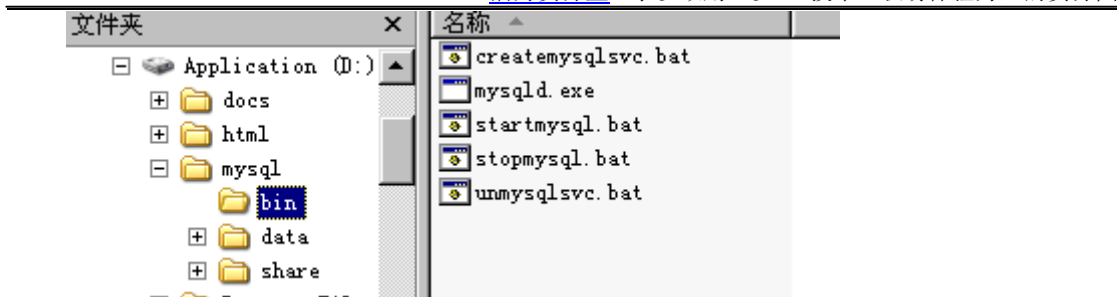
再下载图形管理工具，如下图：



(图 9-16 Mysql 下载)

二、安装

1、MySQL5.1 绿色版，解压到 D 盘，解压后目录如下图所示，如不一致，请调整：

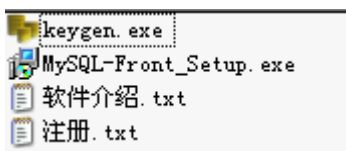


(图 9-17 MySQL 目录)

绿色版的数据文件默认地址为 D: \mysql，如需更改，请修改 mysql 目录下 my.ini 文件的 basedir 和 datadir 两项，注意 MySQL 只支持绝对路径。

要运行 MySQL，先双击 createmysqlsvc.bat 将 mysql 注册成服务，在电脑重启后将自动启动，再双击 startmysql.bat 即启动服务，停止服务双击 stopmysql.bat。如果你不需要 MySQL 随着电脑的启动而自动运行，请运行 unmysqlsvc.bat。详情可参考 18.2 系统安装。

2、管理界面，解压 mysql-front，解压后文件如下：



双击 MySQL-Front_Setup.exe，安装时，一路默认即可。

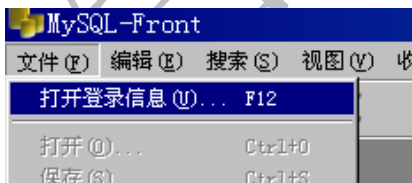
三、运行管理界面，并连接到 MySQL



1、启动 mysql 服务，双击 MySQL-Front 图标，运行，第一次运行时，会弹出两个对话框，直接点取消。

接着选择菜单里的 帮助 -> 登记 -> 要求输入 Key，此时输入注册码(在文件“注册.txt”中)即可。

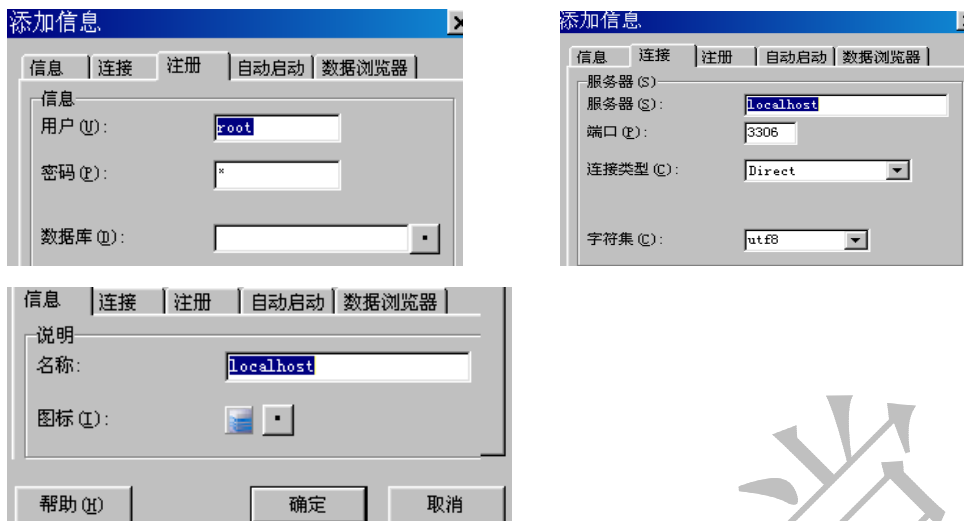
2、文件->打开登录信息



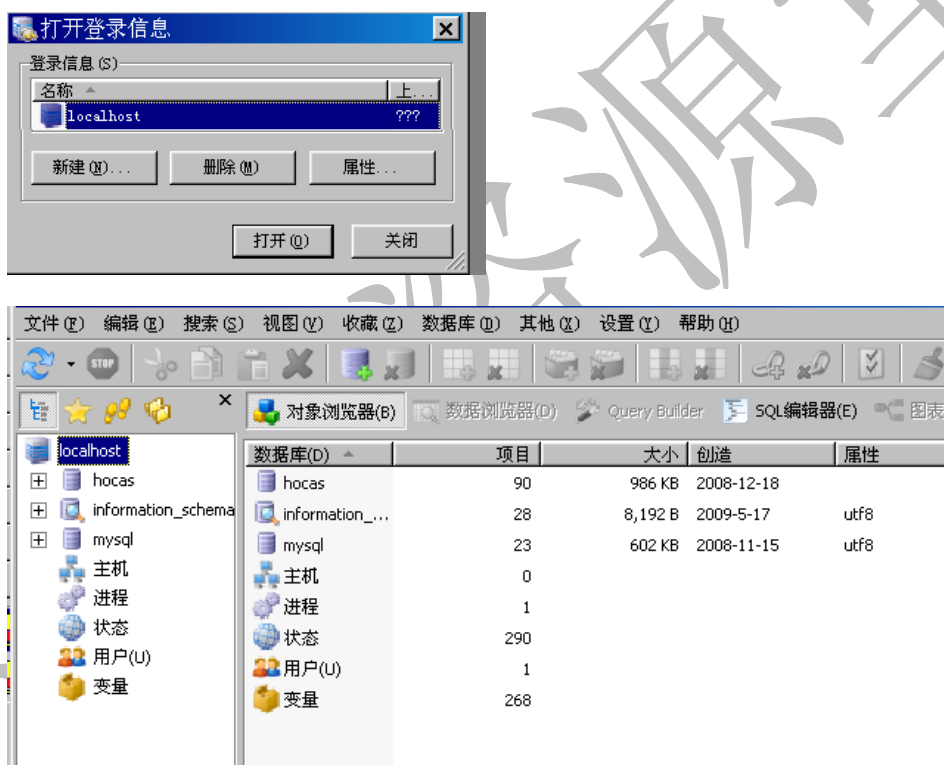
弹出添加信息对话框：

注册：用户名：root 密码：1

连接：服务器：localhost 字符集：utf8，其它默认



点确定，弹出打开登录信息对话框，点打开，则得到如下界面。

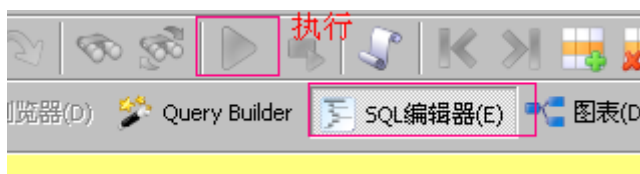


(图 9-17 MySQL-Front 管理界面)

此时，已成功连接至 MySQL 服务器。

四、使用 sql，进行基本操作

点击 SQL 编辑器，得到如下界面：



1、创建数据库


```
create database test;
```

2、选择使用该数据库

```
use test;
```

3、建表

Student 表:

```
create table student(
    id int primary key,
    name varchar(20)
);
```

一个学生，一张 IdCard

IdCard 表:

```
create table idcard(
    id int primary key,
    studentid int ,
    cardInfo varchar(50) not null,
    foreign key(studentid) references student(id)
);
```

```
1 create table student(
2 id int primary key,
3 name varchar(20)
4 );
5
```

Field	Type	Null	Key
id	<MEMO>	NO	PRI
name	<MEMO>	YES	

```
1 create table idcard(
2 id int primary key,
3 studentid int ,
4 cardInfo varchar(50) not null,
5 foreign key(studentid) references student(id)
6 );
```

Field	Type	Null	Key	Default	Extra
id	<MEMO>	NO	PRI		
studentid	<MEMO>	YES	MUL		
cardInfo	<MEMO>	NO			

4、增删改查操作

增:

```
insert into student values(0,'ad');
```

```
insert into student values(1,'ad1');
```

查:

```
select * from student;
```

```
1 select * from student;
```

id	name
0	ad
1	ad1

改:

```
update student set name = 'ad_update' where id = 1
```

执行完后表记录

id	name
0	ad
1	ad_update

删: delete from student where id = 1

执行完后，表记录

id	name
0	ad

5、mysql 所支持的自动递增

一个表只能有一个 AUTO_INCREMENT 属性，且该属性必须为主键的一部分

```
create table test1(id int primary key AUTO_INCREMENT,name varchar(20));
```

6、mysql 所支持的分页处理

先在 student 表中添加数据：

```
insert into student values(1,'ad1');
```

```
insert into student values(2,'ad2');
```

```
insert into student values(3,'ad3');
```

```
insert into student values(4,'ad4');
```

```
insert into student values(5,'ad5');
```

查看表中数据用于对比

id	name
0	ad
1	ad1
2	ad2
3	ad3
4	ad4
5	ad5

使用 limit 关键字：

`select * from student limit 2,3;`（从第 2 条记录开始，取出 3 条数据，注意数据库中记录从 0 开始算）

id	name
2	ad2
3	ad3
4	ad4

第 10 章 了解 JSP

JSP（JavaServer Pages）是由 Sun 公司倡导、众多公司参与建立的动态网页技术，以 Java 语言作为脚本。JSP 能够快速开发出基于 Web、独立于平台的应用程序，用户界面与系统内容分离，使得设计人员能够在不改变底层动态内容的前提下改变整个网页布局。

通过本章的学习，能初步掌握 JSP 开发的基本知识。

对语法的学习，一般只需要先了解有那些基本的语法，然后在使用中实践并验证这些语法，并不断学习新的语法。切不可先完全搞懂语法才去实践，这样只会浪费时间。

最常用的 JSP 语法就以下几个：include、import、contentType、脚本元素和变量、param、useBean 等。

总之，你得尽快从一个又一个的页面先做起来，哪怕是最简单的页面。当你看到页面的效果，自豪感油然而生……

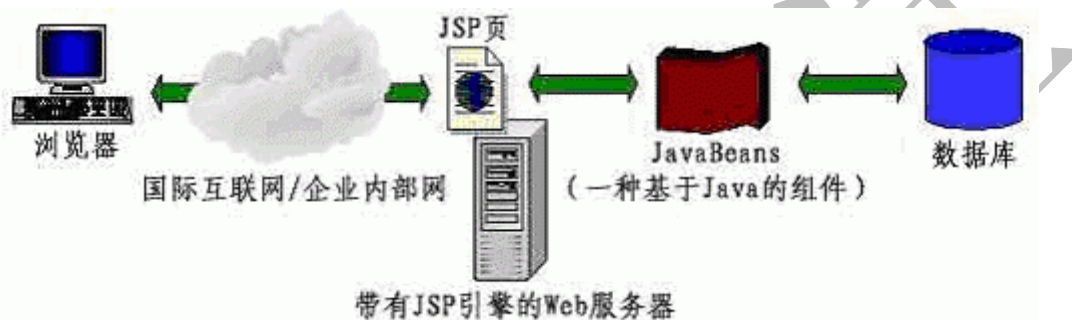
这样，你很快就入门了。

10.1 初识 JSP

JSP 是建立在 Java Servlets 模型之上的表现层 (UI) 技术，使 HTML 的编写变得更简单。允许将静态 HTML 内容与服务器端脚本混合起来动态输出。JSP 为服务端的 Java 库提供了接口，可以随时响应客户端请求，而动态生成 HTML、XML 或其他格式的文档。

JSP 请求模型

现在来看看 JSP 是如何处理 HTTP 请求的。在基本请求模型中，一个请求直接被送到 JSP 页中。图 10-1 说明了这个模型中的信息流程。JSP 代码控制着进行逻辑处理时与 JavaBeans 组件的交互，并在动态生成的、混合了静态 HTML 代码的 HTML 页中显示结果。



(图 10-1 JSP 基本请求模型)

Beans 可以是 JavaBeans、EJB 组件等，复杂的请求模型可从被请求页呼叫其他 JSP 页、Java Servlets、URL 地址等。

JSP 引擎技术内幕

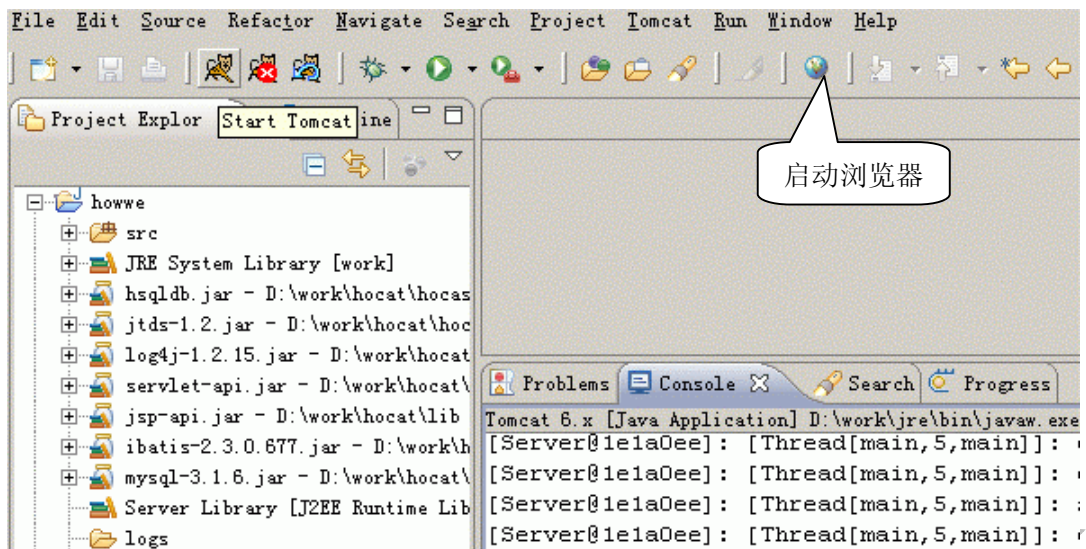
JSP 引擎实际上是把 JSP 标签、JSP 页中的 Java 代码甚至连同静态 HTML 内容都转换为大块的 Java 代码。所有代码被 JSP 引擎生成 Servlet，然后再自动编译成 Java 字节码。



当网站的访问者请求一个 JSP 页时，其实是一个已经生成的、预编译过的 servlet 在完成所有工作。这也是 JSP 比 ASP 高效的原因。因为 servlet 是编译过的，所以网页中的 JSP 代码不需要在每次请求该页时被解释。JSP 引擎只需在 servlet 代码最后被修改后编译一次，然后这个编译过的 servlet 就可以被执行了。由于是 JSP 引擎自动生成并编译 servlet，不用程序员动手编译代码，所以 JSP 能带给你高效的性能和快速开发所需的灵活性。

10.2 JSP 版 Hello World

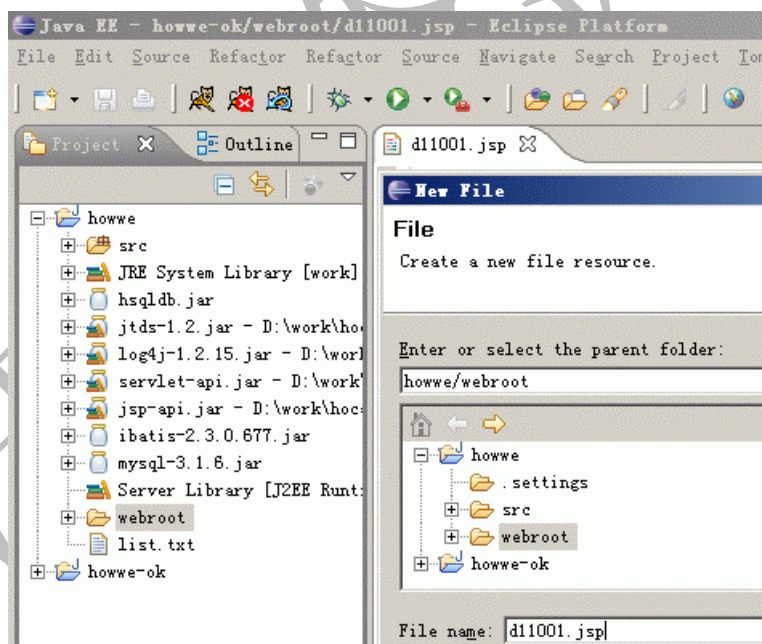
我们先启动 Web 服务器，在 1.3 节中已经讲述了三种启动方式，第 4 种是直接在 Eclipse 中启动。如图 10-2 所示，单击图中提示 Start Tomcat 的图标，即开始启动 Web 服务器。单击图中提示 启动浏览器 的图标，启动一浏览器窗口。



(图 10-2 在 Eclipse 中启动 tomcat)

注意：如果已经将 hocat 注册成服务，请先取消服务。如果已经启动了 Web 服务器，再次启动无效，因为在同一端口没法绑定两个服务，原因请参考“第 1 章 小知识：什么是服务”。

我们开始做一个简单的 Jsp 页面。



(图 10-3 在 howwe 项目新建页面)

在项目 howwe 的 webroot 目录上单击右键，然后选择 **New => File**。如图 10-3 所示，在 File name 处键入 d11001.jsp 作为文件名称，然后按 Finish。

在文件中输入如下代码并保存：

```
<HTML>
<BODY>
<%out.println("Hello World"); %>
</BODY>
```


</HTML>



(图 10-4 选择 Jsp 页面链接)

如图 10-4 所示, 在地址栏输入 127.0.0.1, 用鼠标单击选择如图所示的链接, 则出现如图 10-5 所示的网页。当然也可以在地址栏直接输入 <http://127.0.0.1/me/d11001.jsp> 来访问。注意, howwe 项目已经配置成访问地址为 me, 具体的怎么配置请参考《从人生需要引导论 Java 快速开发》“第 19 章 Web 应用工作原理”。



(图 10-5 浏览 Jsp 页面)

10.3 数据库版 Hello World

将第 6 章的范例部分代码转化为 JSP 页面, 这次采用 SERVER 访问模式, 这种模式下数据库的操作和大中型数据库 (如 DB2、SQL SERVER、ORACLE 等) 一样。HSQL 数据库的访问模式请参考“第 8 章 认识开源数据 HSQLDB”。

电话外呼系统 (详情请参考“第 14 章 电话外呼系统”) 使用 HSQL 作为数据库, 数据库名 webapp。表 test 专门用于演示 Hello World, 字段为: cnt (varchar)。

在项目 howwe 的 webroot 目录上单击右键, 然后选择 New => File, 为文件名称为 d11003r.jsp, 然后按 Finish。

在文件中输入如下代码并保存:

```
<%@ page contentType="text/html; charset=gbk"%>
<%
java.sql.Connection conn = null;    //数据库连接
java.sql.Statement st =null;        //执行语句对象
```



```
java.sql.ResultSet rs=null;           //记录集
String sSql=null;
try {
    //创建数据库连接
    Class.forName("org.hsqldb.jdbcDriver");    //加载数据库驱动
    conn=java.sql.DriverManager.getConnection("jdbc:hsqldb:hsqldb://127.0.0.1:19132/webapp", "sa", "sa1");    //数据库连接，采用SERVER访问模式
    st=conn.createStatement();    //创建执行语句对象

    //查询记录
    sSql="SELECT cnt FROM test";
    rs = st.executeQuery(sSql);
    System.out.println("读取记录（英文内容）");
}%>

<HTML>
<BODY>
<%
    while(rs.next())
    {
        数据库内容为: <%=rs.getString(1)%><BR>
    }
}%>
} catch (java.sql.SQLException e) {
    e.printStackTrace();    //显示数据操作出错原因
} catch (ClassNotFoundException e) {
    e.printStackTrace();    //不存在org.hsqldb.jdbcDriver时出错
}
}%>
</BODY>
</HTML>
```



(图 10-6 浏览 Jsp 页面-读数据库)

10.4 HTML 语法

HTML 是 Hypertext Markup Language 的英文缩写，即超文本标记语言。是一种用来制

作网页的标记语言，不需要编译，直接由浏览器执行。对大小写不敏感，HTML 与 html 是一样的。由 W3C 维护，常用标准版本为 1999 年 12 月的 HTML 4.01。

HTML 文件是一个文本文件，包含了一些 HTML 元素、标签等。HTML 文件必须使用 html 或 htm 为文件名后缀。

一、HTML 文件框架

文件框架的作用是描述 HTML 文件的结构，但 HTML 中有一种元素就叫框架 (Frame)

HTML 范例: d11004html.htm

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html>
  <head>
    <title>HTML文件标题</title>
    <!-- HTML头信息 -->
  </head>

  <body>
    HTML内容信息
  </body>
</html>
```

HTML 文档首先要声明一个文档类型，也就是例子中的第一行，定义了 XHTML1.1 文档类型 (文档类型是为了说明这个页面使用了何种 HTML 规则或者结构，一般可以省略)。

<html>和</html>是 HTML 文档的开始与结束，也是 HTML 文档的根元素。除了文档类型外的所有页面内容，都包括在 html 元素。

HTML 文件中主要分为头信息 head 与内容信息 body:

1、文档 head 信息

头信息 head 可以容纳文档的 HTML 相关信息，比如标题 title，页面的语言与文字类型，css 样式，javascript 代码，简短描述，关键词等内容，是用户无法直接看到的。

2、文档 body 信息

内容信息 body 包括用户可以看到的全部内容，比如段落，链接，表格等。

注意：编写代码时一定要有层次感，使用 tab 键产生 HTML 代码的层次感，可以使代码更清晰，更容易扩展。不过 tab 键在不同编辑器上所占的空格不同，有的为 4 个，有的为 8 个，建议用 4 个空格代替 tab 键。

二、常用 html 标签

1、html 框架类标签

```
html -- 定义 html 文档
body -- 定义文档体 body
head -- 定义文档头信息 head
title -- 定义文档的标题 title
<!-- ... --> -- html 注释标签
```

2、html 图像与链接类标签

a -- html 链接标签 a

img -- html 图像标签 img

html 文字相关标签

h -- 定义标题 1 至标题 6 h1, h2, h3, h4, h5, h6

p -- html 段落标签 p

div -- html 层 div

strong -- 定义要强调显示的内容 strong

3、html 列表标签

ul -- 定义 html 列表 ul

li -- 定义 html 列表内容 li

4、html 表格类标签

table -- 定义 html 表格 table

tr -- 定义表格行 tr

td -- 定义表格列 td

5、html 表单类标签

form -- html 表单标签 form

input -- 定义一个表单的输入域 input

select -- 定义可选择的 html 表单 select

textarea -- 定义一个多行的文字输入域 textarea

以上 20 个标签，如果能全部掌握，网页中 95% 的 html 代码就掌握了。如果遇到生疏的 html 代码，可以 Google 一把。

html 标签主要分两类：两边封闭（如 html）、自封闭。

自封闭的 html 标签，非成对出现（单独出现），而且是以 /> 结尾，称为自封闭。所有自封闭的 html 标签如下：

br、hr、col、img、area、base、link、meta、frame、input、param、DOCTYPE、isindex、basefont。

三、HTML 属性

HTML 属性一般都出现在 HTML 标签中，HTML 属性是 HTML 标签的一部分。

标签可以有属性，它包含了额外的信息。属性的值一定要在双引号中。标签可以拥有多个属性，属性由属性名和值成对出现。

HTML 属性语法

<标签名 属性名 1="属性值" 属性名 2="属性值" ... 属性名 N="属性值"></标签名>

范例：

标签<a>是超链接标签，使用 href 属性，可以定义链接的位置 (URI) 代码。

四、HTML 元素

HTML 元素是构建网页的一种单位，是由 HTML 标签和 HTML 属性组成的，HTML 元素也是网页中的一种基本单位。

范例：

```
<a href="http://howwe.net">人生需要引导</a>
```

这是一个 HTML 链接元素。

```
<p>这是我的第一个网页，来吧
```

```
  <a href="http://test.howwe.net">尽情学习</a>吧！
```

```
</p>
```

这是一个 HTML 段落元素，它包含了一个 HTML 链接元素。

五、HTML 文档

HTML 文档就是 HTML 页面，也就是网页，是由 HTML 元素组成。互联网所有内容都是由一个个 HTML 文档展现。

六、HTML 注释

HTML 注释，我们经常要在一些代码旁做一些 HTML 注释。这样做的好处很多：方便查找、方便比对、方便项目组里的其它程序员了解你的代码，而且可以方便以后你對自己代码的理解与修改等。

HTML 注释的开始使用<!--，结束使用-->

HTML 注释语法

```
<!--注释的内容-->
```

10.5 快速掌握 JSP

一、语法

一个 JSP 页面可以被分为以下几部份：

- 静态数据，如 HTML
- JSP 指令，如 include 指令
- JSP 脚本元素和变量
- JSP 动作
- 用户自定义标签

静态数据

静态数据就是不变的数据，在文件中的内容和输出给 HTTP 响应的内容完全一致。此时，该 JSP 输入文件会是一个没有内嵌 Java 或动作的 HTML 页面。而且，客户端每次请求都会得到相同的响应内容。

JSP 指令

JSP 指令控制 JSP 编译器如何去生成 servlet，以下是可用的指令：

* 包含指令 include - 包含指令通知 JSP 编译器把另外一个文件完全包含入当前文件中。效果就好象被包含文件的内容直接被粘贴到当前文件中一样。这个功能和 C preprocessor 所提供的很类似。被包含文件的扩展名一般都是“jspf”（即 JSP Fragment，JSP 碎片）：

```
<%@ include file="somefile.jsp" %>
```

* 页面指令 page - 页面指令有以下几个选项:

import 使一个 Java 导入声明被插入到最终页面文件。

contentType 规定了生成内容的类型。当生成非 HTML 内容或者当前字符集 character set 并非默认字符集时使用。

errorPage 处理 HTTP 请求时, 如果出现异常则显示该错误提示信息页面。

isErrorPage 如果设置为 TRUE, 则表示当前文件是一个错误提示页面。

isThreadSafe 表示最终生成的 servlet 是否安全线程 (thread safe)。

```
<%@ page import="java.util.*" %> //import 导入样例, 多个可用逗号 (,) 隔开
```

```
<%@ page contentType="text/html" %> //contentType 页面类型样例
```

```
<%@ page isErrorPage=false %> //for non error page 无错页面样例
```

```
<%@ page isThreadSafe=true %> //for a thread safe JSP 安全线程 JSP 样例
```

注意: 在同一个 JSP 文件中只有 "import" 导入页面指令可以被多次使用。

* 标签库指令 taglib - 标签库指令描述了要使用的 JSP 标签库。该指令需要指定一个前缀 prefix (和 C++ 的命名空间很类似) 和标签库的描述 URI:

```
<%@ taglib prefix="myprefix" uri="taglib/mytag.tld" %>
```

JSP 脚本元素和变量

标准脚本变量, 以下是永远可用的脚本变量:

* out - JSPWriter 用来写入响应流的数据

* page - servlet 自身

* pageContext - 一个 PageContext 实例包括和整个页面相联系的数据, 一个给定的 HTML 页面可以在多个 JSP 之间传递

* request - HTTP request 对象

* response - HTTP response 对象

* session - HTTP session 对象, 在多个请求之间追踪一个客户的信息。

脚本元素, 有三个基本的脚本元素, 作用是使 Java 代码可以直接插入 servlet.

* 一种是脚本标签, 在 Java SERVLET 的类的 _jspService() 方法中放入所包含的语句。

```
<% int localStackBasedVariable = 1; out.println(localStackBasedVariable); %>
```

* 一种是声明标签, 在 Java SERVLET 的类体中放入一个变量的定义。静态的数据成员也可以如此定义, 在实际中很少使用。

```
<%! int serverInstanceVariable = 1; %>
```

* 一种是表达式标签, 在 Java SERVLET 的类中放入待赋值的表达式。

```
<%= "expanded inline data " + 1 %>
```

注意: 表达式不能以分号结尾。

JSP 动作

JSP 动作是一系列可以调用内建于网络服务器中的功能的 XML 标签。JSP 提供了以下动作:

jsp:include 和子过程类似, Java SERVLET 暂时接管对其它指定的 JSP 页的请求

和响应。当处理完该 JSP 页后就马上把控制权交还当前 JSP 页。这样 JSP 代码就可以在多个 JSP 页中共享而不用复制。

`jsp:param` 可以在 `jsp:include`, `jsp:forward` 或 `jsp:params` 块之间使用。指定一个将加入请求的当前参数组中的参数。

`jsp:forward` 用于处理对另一个 JSP 或 SERVLET 的请求和响应。控制权永远不会交还给当前 JSP 页。

`jsp:plugin` Netscape Navigator 的老版本和 Internet Explorer 使用不同的标签以嵌入一个 applet。这个动作产生为嵌入一个 APPLET 所需要的指定浏览器标签。

`jsp:fallback` 如果浏览器不支持 APPLETS 则会显示的内容。

`jsp:getProperty` 从指定的 JavaBean 中得到一个属性值。

`jsp:setProperty` 在指定的 JavaBean 中设置一个属性值。

`jsp:useBean` 创建或者复用一个 JavaBean 变量到 JSP 页。

JSP 标签库

除了 JSP 预定义动作之外,开发者还可以使用 JSP 标签扩展 API 添加他们自定义的动作。开发者写一种实现一个标签的界面和一个标签库的 XML 描述文件的 Java 类,这就能指定标签和实现标签的 Java 类。使用如下:

```
<%@ taglib uri="mytaglib.tld" prefix="myprefix" %>
```

```
<myprefix:myaction> <!-- the start tag %>
```

```
...
```

```
</myprefix:myaction> <!-- the end tag %>
```

以上是比较常用的语法。对语法的学习,一般只需要先了解有那些基本的语法,然后在使用中实践并验证这些语法,并不断学习新的语法。切不可先完全搞懂语法才去实践,这样只会浪费时间。

最常用的就以下几个: `include`、`import`、`contentType`、脚本元素和变量、`param`、`useBean` 等。

总之,你得尽快从一个又一个的页面先做起来,哪怕是最简单的页面。当你看到页面的效果,自豪感油然而生……

二、JSP 四大对象及范围

在 JSP 页面中的对象,包括用户创建的对象(如 JavaBean 对象)和 JSP 的隐含对象,都有一个范围属性。范围定义了在规定时间内,在哪个 JSP 页面中可以访问这些对象。例如, `session` 对象在会话期间内,可以在多个页面中被访问。`application` 对象在整个 Web 应用程序的生命周期中都可以被访问。在 JSP 中,有 4 种范围,如下所示。

1、page 范围

具有 page 范围的对象被绑定到 `javax.servlet.jsp.PageContext` 对象中。在这个范围内的对象,只能在创建对象的页面中访问。可以调用 `pageContext` 这个隐含对象的 `getAttribute()` 方法来访问具有这种范围类型的对象(`pageContext` 对象还提供了访问其他范围对象的 `getAttribute` 方法),`pageContext` 对象本身也属于 page 范围。当 Servlet 类的 `_jspService()` 方法执行完毕,属于 page 范围的对象引用将被丢弃。

page 范围内的对象，在客户端每次请求 JSP 页面时创建，在页面向客户端发送响应或请求被转发（forward）到其他的资源后被删除。

2、request 范围

具有 request 范围的对象被绑定到 `javax.servlet.HttpServletRequest` 对象中，可以调用 `request` 这个隐含对象的 `getAttribute()` 方法来访问具有这种范围类型的对象。在调用 `forward()` 方法转向的页面或者调用 `include()` 方法包含的页面中，都可以访问这个范围内的对象。要注意的是，因为请求对象对于每一个客户请求都是不同的，所以对于每一个新的请求，都要重新创建和删除这个范围内的对象。

3、session 范围

具有 session 范围的对象被绑定到 `javax.servlet.http.HttpSession` 对象中，可以调用 `session` 这个隐含对象的 `getAttribute()` 方法来访问具有这种范围类型的对象。JSP 容器为每一次会话，创建一个 `HttpSession` 对象，在会话期间，可以访问 `session` 范围内的对象。

4、application 范围

具有 application 范围的对象被绑定到 `javax.servlet.ServletContext` 中，可以调用 `application` 这个隐含对象的 `getAttribute()` 方法来访问具有这种范围类型的对象。在 Web 应用程序运行期间，所有的页面都可以访问在这个范围内的对象。

下面我们通过几个简单的例子，来看一下这 4 种范围对象的应用。

1. 测试 page 范围

test1.jsp

```
<%
pageContext.setAttribute("name", "zhangsan");
out.println("test1.jsp: ");
out.println(pageContext.getAttribute("name"));
out.println("<p>");
pageContext.include("test2.jsp");
%>
```

test2.jsp

```
<%
out.println("test2.jsp: ");
out.println(pageContext.getAttribute("name"));
%>
```

访问 test1.jsp，将看到如下的输出：

```
test1.jsp: zhangsan
test2.jsp: null
```

说明保存在 `pageContext` 对象中的属性具有 page 范围，只能在同一个页面中被访问。

2. 测试 request 范围

修改 test1.jsp 和 test2.jsp，如下所示。

test1.jsp

```
<%
request.setAttribute("name", "zhangsan");
out.println("test1.jsp: ");
out.println(request.getAttribute("name"));
out.println("<p>");
pageContext.include("test2.jsp");
%>
```

test2.jsp

```
<%
out.println("test2.jsp: ");
out.println(request.getAttribute("name"));
%>
```

访问 test1.jsp，将看到如下的输出：

```
test1.jsp: zhangsan
test2.jsp: zhangsan
```

说明保存在 request 对象中的属性具有 request 范围，在请求对象存活期间，可以访问这个范围内的对象。将 `pageContext.include("test2.jsp");` 这一句注释起来，先访问 test1.jsp，再访问 test2.jsp，可以看到如下输出：

```
test2.jsp: null
```

这是因为客户端开始了一个新的请求。

3. 测试 session 范围

修改 test1.jsp 和 test2.jsp，如下所示。

test1.jsp

```
<%
session.setAttribute("name", "zhangsan");
%>
```

test2.jsp

```
<%
out.println("test2.jsp: ");
out.println(session.getAttribute("name"));
%>
```

先访问 test1.jsp，然后在同一个浏览器窗口中访问 test2.jsp，可以看到如下输出：

```
test2.jsp: zhangsan
```

说明保存在 session 对象中的属性具有 session 范围，在会话期间，可以访问这个范围

内的对象。

如果我们在访问完 test1.jsp 后，关闭浏览器，重新打开浏览器窗口，访问 test2.jsp，将看到如下输出：

test2.jsp: null

这是因为客户端与服务器开始了一次新的会话。

4. 测试 application 范围

修改 test1.jsp 和 test2.jsp，如下所示。

test1.jsp

```
<%  
application.setAttribute("name", "zhangsan");  
%>
```

test2.jsp

```
<%  
out.println("test2.jsp: ");  
out.println(application.getAttribute("name"));  
%>
```

先访问 test1.jsp，然后关闭浏览器，再打开浏览器窗口，访问 test2.jsp，可以看到如下输出：

test2.jsp: zhangsan

说明保存在 application 对象中的属性具有 application 范围，在 Web 应用程序运行期间，都可以访问这个范围内的对象。

三、Jsp 页面由浅入深

下面示范几个 jsp 页面，由浅入深，迅速了解 JSP。

1、dl10test01.jsp

```
<HTML>  
<BODY>  
Hello! The time is now <%= new java.util.Date() %>  
</BODY>  
</HTML>
```

每次在浏览器中重载网页时，它就出现当前时间。字符系列<%= and %>的作用是引入 Java 表达式，这个表达式将在运行的时候被计算。正因为这样，使用 JSP 产生动态 HTML 网页来响应用户的动作才变为可能。

2、dl10test02.jsp

```
<HTML>  
<BODY>  
<%  
System.out.println( "Evaluating date now" );  
java.util.Date date = new java.util.Date();
```

```
%>
```

```
Hello! The time is now <%= date %>
</BODY>
</HTML>
```

编写 Java 代码块嵌入到 JSP 中。具体做法是：将 Java 代码段放置于<% and %>字符之间。其实跟表达式是一样的，只不过在开始的地方没有等号而已。

3、dl10test03.jsp

```
<%@ page import="java.util.Date" %>
<HTML>
<BODY>
<%!
Date theDate = new Date();

Date getDate()
{
    System.out.println( "In getDate() method" );
    return theDate;
}
%>
```

```
Hello! The time is now <%= getDate() %>
</BODY>
</HTML>
```

```
Hello! The time is now <%= getDate() %>
</BODY>
</HTML>
```

为了增加一个声明，必须使用<%! and %>来包含声明。dl10test03.jsp 已经声明了变量和方法：一个 Date 变量 theDate 以及一个方法 getDate。

但是这个例子并不能正常工作：不管你怎么重载网页，日期都是相同的。原因是这些声明，只在网页被装载的时候才被计算一次。这一点就象我们在 Visual C++中创建一个类并定义变量的初始化值。

4、dl10test04.jsp

```
<%@ page import="java.util.Date" %>
<HTML>
<BODY>
<%
System.out.println( "Evaluating date now" );
Date date = new Date();
```


%>

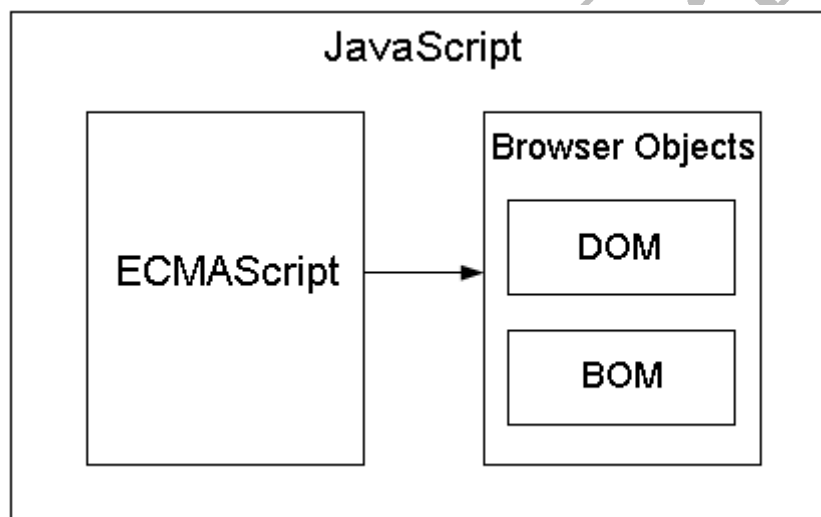
```
Hello! The time is now <%= date %>
</BODY>
</HTML>
```

这种方式比较常用，条理清晰。

10.6 JavaScript 语法

一、简介

JavaScript，简称为 js，是一种基于对象和事件驱动的客户端脚本语言，起源于 Netscape 公司的 LiveScript 语言。最初是为了检验 HTML 表单输入的正确性。



(图 10-7 js 组成)

如图 10-7 所示，JavaScript 是由 ECMAScript（语法）、Browser Objects（DOM、BOM）（特性）组成，可以检测表单的正确性，实现 Ajax，读、写、改变 HTML 页面的架构 DOM，对事件做出响应，检测浏览器所使用的设备，产生很酷很炫的网页效果 DHTML 等。

Javascript 与 Java 没有任何关系，他们本身就是两种语言，取成差不多的名字，主要是由于商业上的原因。Javascript 是一种客户端脚本语言，Java 是服务器端语言。

二、JavaScript 相关知识

以下是 JavaScript 相关知识点的列表，从中可以看出，JavaScript 也包含数据类型、控制结构、数据结构三部分。

1. 在 HTML 中嵌入 JavaScript
 - 1.1. 内部引用 JavaScript
 - 1.2. 外部引用 JavaScript
 - 1.3. 内联引用 JavaScript
2. JavaScript 语法
 - 2.1. JavaScript 注释
 - 2.2. JavaScript 变量

- 2.3. JavaScript 数值类型
- 2.4. JavaScript 字符串类型
- 2.5. JavaScript 运算符与表达式
 - 2.5.1. JavaScript 算术运算符与表达式
 - 2.5.2. JavaScript 赋值运算符与表达式
 - 2.5.3. JavaScript 自增、自减运算符与表达式
 - 2.5.4. JavaScript 逗号运算符与表达式
- 2.6. JavaScript 程序设计
- 2.7. JavaScript 顺序程序设计
- 2.8. JavaScript 选择程序设计
 - 2.8.1. JavaScript 布尔类型
 - 2.8.2. JavaScript 关系运算符与表达式
 - 2.8.3. JavaScript 逻辑运算符与表达式
 - 2.8.4. JavaScript if...else 语句
 - 2.8.5. JavaScript 条件运算符与表达式
 - 2.8.6. JavaScript switch...case 语句
- 2.9. JavaScript 循环程序设计
 - 2.9.1. JavaScript while 语句
 - 2.9.2. JavaScript do...while 语句
 - 2.9.3. JavaScript for 语句
 - 2.9.4. JavaScript break 与 continue 语句
- 3. JavaScript 面向对象 (OOP) 语法
 - 3.1. JavaScript 面向对象代码实践
 - 3.2. 使用构造函数创建 JavaScript 对象
 - 3.3. JSON 法创建 JavaScript 对象

三、JavaScript 的嵌入方式:

1、内部引用：通过 HTML 的 script 标签加载 JavaScript 代码。

范例

```
<head>
<script type="text/javascript">
    document.write("welcome!");
</script>
</head>
```

可以通过注释隐藏 JavaScript 代码

上面范例可改为:

```
<head>
<script type="text/javascript">
<!--
    document.write("welcome!");
//-->
</script>
</head>
```

<!-- ... //-->当浏览器不支持 JavaScript 时，屏蔽 JavaScript 代码。

现在这种隐藏 JavaScript 代码的方式可以忽略，因为没有浏览器不支持 JavaScript，除了用户手动禁止浏览器的 JavaScript 功能，但是这种情况很少发生。

注意：在 XHTML 文档中使用 CDATA 避免解析 HTML 实体，即使用 CDATA 囊括的内容中的 HTML 实体将不被翻译。具体内容请 Google。

高级用法：使用 noscript 标签为用户提供更好的体验

通过 JavaScript 注释的方式可以隐藏 JavaScript 代码，通过 noscript 标签可以为用户提供更好的体验（提示用户你的浏览器不支持 JavaScript）

示例

```
<body>
  <script type="text/javascript">
    document.write("welcome!");
  </script>
  <noscript>
    <p>你浏览器的 JavaScript 功能已被禁用，或不支持 JavaScript。请更改浏览器选项以启用 JavaScript。
  </p>
  </noscript>
</body>
```

2、外部引用：引用 HTML 文件外部的 JavaScript 文件。

这种方式可以使代码更清晰，更容易扩展。通过 HTML 的 script 标签加载 JavaScript 文件。

标准方法是把 JavaScript 文件放到 head 标签内，JavaScript 文件必须使用 js 为文件后缀。

```
<head>
  <script type="text/javascript" src="hwui.js"></script>
</head>
```

为防止网页加载缓慢，也可以把非关键的 JavaScript 放到网页底部。

外部引用的优点：统一定义 JavaScript 代码，方便查看，方便维护；使代码更安全，可以压缩，加密单个 JavaScript 文件；浏览器可以缓存 JavaScript 文件，减少宽带使用（当多个页面同时使用一个 JavaScript 文件的时候，通常只需下载一次）。

注意：尽量不要把 JavaScript 分为多个文件，多个文件会增加服务器的负担，增加服务器的 HTTP 请求。

3、内联引用：是通过 HTML 标签中的事件属性实现的。

范例：

```
<input type="button" value="测试" onclick="alert('你点击了一个按钮');">
```

上面示例将调用 input 标签的 onclick 属性，弹出一个提示框。

10.7 CSS 语法

CSS，Cascading Style Sheets 的缩写，中文名**层叠样式表**。CSS 语言是一种标记语言，不需要编译，可以直接由浏览器执行（属于浏览器解释型语言）。

CSS 用于网页布局及网页美化。可以通过简单更改 CSS 文件，而改变网页的整体表现形式。这样可以减少我们的工作量，所以 CSS 是每一个网页设计人员的必修课。

CSS 文件是文本文件，它包含 CSS 标记。CSS 文件必须使用 `css` 为文件名后缀，对大小写不敏感，由 W3C 的 CSS 工作组产生和维护。

CSS 语法非常简单，组成 CSS 语法的元素只有 CSS 选择符与 CSS 属性。每个 CSS 选择符由 1 个或多个 CSS 属性组成。

以下是 CSS 相关知识点的列表：

1. CSS 语法 -- 最基本的
 - 1.1. 外部引用 CSS
 - 1.1.1. 使用 `link` 标签引用 CSS
 - 1.2. 内部引用 CSS
 - 1.3. 内联引用 CSS
 - 1.4. CSS 选择符
 - 1.5. CSS 声明
 - 1.6. CSS 注释
2. CSS `color` 属性 -- CSS 前景色
 - 2.1. CSS `opacity` 属性
3. CSS 颜色 -- 五颜六色的世界
 - 3.1. CSS `RGB(A)` 颜色
 - 3.2. CSS `HSL(A)` 颜色
 - 3.3. 短 16 进制颜色与 web 安全色
4. CSS 背景
 - 4.1. CSS `background-color` 属性
 - 4.2. CSS `background-image` 属性
 - 4.3. CSS `background-repeat` 属性
 - 4.4. CSS `background-position` 属性
 - 4.5. CSS `background-attachment` 属性
 - 4.6. CSS `background` 属性
5. CSS 文本
 - 5.1. CSS `letter-spacing` 属性
 - 5.2. CSS `word-spacing` 属性
 - 5.3. CSS `text-decoration` 属性
 - 5.4. CSS `text-transform` 属性
 - 5.5. CSS `text-align` 属性
 - 5.6. CSS `text-indent` 属性

- 5.7. CSS white-space 属性
- 6. CSS 字体
 - 6.1. CSS font-family 属性
 - 6.1.1. CSS family-name 系列性字体名称
 - 6.1.2. CSS generic-family 一般性字体名称
 - 6.2. CSS font-size 属性
 - 6.3. CSS font-style 属性
 - 6.4. CSS font-variant 属性
 - 6.5. CSS font-weight 属性
 - 6.6. CSS font 属性
- 7. CSS 列表
 - 7.1. CSS list-style-type 属性
 - 7.2. CSS list-style-image 属性
 - 7.3. CSS list-style-position 属性
 - 7.4. CSS list-style 属性
- 8. CSS cursor 属性 -- 鼠标样式
- 9. CSS 边框
 - 9.1. CSS border-width 属性
 - 9.2. CSS border-color 属性
 - 9.3. CSS border-style 属性
 - 9.4. CSS border 属性
- 10. CSS margin 属性 -- CSS 边外补白
- 11. CSS padding 属性 -- CSS 边内补白

下面简单介绍 CSS 相关知识点:

- 1、使用 link 标签引用 CSS (注意: style 标签应该在 head 标签内部)

范例:

```
<head>
  <link rel="stylesheet" type="text/css" href="http://www.howwe.net/style.css" />
</head>
```

- 2、内部引用 CSS, 使用 style 标签直接把 CSS 文件中的内容加载到 HTML 文档内部。

范例:

```
<style type="text/css"><!--
/* -----文字样式开始----- */

/* 说明一 */
.whitel2px
{
    color:white;
    font-size:12px;
}

/* 说明二 */
.black16px
```



```
{
    color:black;
    font-size:16px;
}

/* -----文字样式结束----- */
-->
</style>
```

注意：style 标签应该在 head 标签内部。

3、内联引用可以把 CSS 样式直接作用在 HTML 标签中。

范例：

```
<p style="font-size: 10px; color: #FFFFFF;">
    使用 CSS 内联引用表现段落.
</p>
```

4、CSS 选择符 -- CSS 的名字

CSS 选择符就是 CSS 样式的名字，当在 HTML 文档中表现一个 CSS 样式的时候，就要用到一个 CSS。怎么用呢？这时就通过 CSS 选择符(CSS 的名字)来指定此 HTML 标签使用此 CSS 样式。

选择符语法：

选择符名字

```
{
    声明;
}
```

5、选择符取名规则：

CSS 选择符必须以字母开头，使用英文字母的大写与小写、数字、连字号、下划线、冒号、句号。

常用的选择符有三种：

xhtml 标签选择符，比如 p 标签选择符(代表所有的段落都使用这个 CSS 样式)，如 p{font-size:12px;}。

id 选择符，也叫唯一性选择符。如 #red{color:red;}，在名字前增加了一个#。id 选择符在一个页面中只能出现一次，在整个网站中也最好出现一次(这样有利于程序员控制此元素，有多个一样名称的元素，就无法分开不好控制了)。

class 选择符，多重选择符。如.blue{color:blue;}，就是在名字前增加了一个点号(.)，class 选择符在一个页面中可以出现多次。

选择符用法总结

	CSS 中的写法	XHTML 中的写法
xhtml 标签选择符	p{font-size:12px;}	<p>www.howwe.net</p>
id 选择符	#id_selector{font-size:12px;}	<p id="id_selector">梦之都</p>

class 选择符 .class_selector{font-size:12px;} <p class="class_selector">
梦之都</p>

6、CSS 声明是由“属性”，“冒号(:)”，“属性值”和“分号(;)”组成的。

语法：

属性:属性值;

范例

font-size:12px;

font-size 代表字体大小.

12px 字体大小的值.

7、两个常用的技巧

- 选择符的名字一样，声明可以组合;
- 声明全部一样，选择符的名字可以组合。

10.8 编程更需要引导：学以致用

一、有人提出了成为一个 JSP 程序员的基本要求

1、html 掌握程度：熟练。原因：不会 html 你能写 JSP?

2、JavaScript：掌握程度：熟悉。原因：client 端的数据校验、一些页面处理需要你使用脚本。

3、css 掌握程度：熟悉。原因：实现页面风格的统一通常会使用 css 去实现。

4、Java 基础编程掌握程度：熟练。原因：不会 Java 你能写 JSP? 开玩笑吧。还有你必须非常熟悉以下几个包 java.lang; java.io; java.sql; java.util; java.text; javax.srvlet; javax.srvlet.http; javax.mail 等。

5、sql 掌握程度：熟练。原因：如果你不使用数据库的话，你不需要掌握 sql。同时你必须对以下几种数据库中的一种以上的 sql 比较熟悉：Oracle, DB2, Mysql, MsSQL。

6、xml 掌握程度：了解。原因：AppServer 的配置一般是使用 XML 来实现的。

7、ejb 掌握程度：了解。原因：很多项目中商业逻辑是由 ejb 来实现的。

8、以下几种 AppServer 你需要了解一个以上：Tomcat、WebLogic、WebSphere、Jrun、Resin。原因：你的 Jsp 必须跑在 AppServer 上!

9、工具的使用：UltraEdit (EditPlus) +jakarta-ant+jakarta-log4j 或 Jubilder 或 Eclipse 或 Netbeans。

二、按以上要求所需学习时间

1、html 学习时间，如果你的智商在 80 以上，15 天时间应该够用了。至少你能手写出一个页面来。

2、javascript 学习时间，这真的不好说，比较深奥的东西，够用的话一个礼拜可以学写皮毛。

3、css 学习时间，三天的时间你应该知道如何使用 css 了，不要求你写，一般是美工来写 css。

4、Java 学习时间，天才也得三个月吧。慢慢学吧。如果要精通，那我不知道需要多少时间了。用来写 jsp，四个月应该够了。

5、sql 学习时间，只需要知道 insert, delete, update, select, create/drop table 的话一天你应该就知道了。

6、xml 学习时间，很快。

7、ejb 学习时间，基本调用看 3 天你会了。不过是建立在你已经学会 Java 的基础上。

8、熟悉 AppServer, Tomcat 四天你可以掌握安装，配置，把 Jsp 跑起来。

9、Struts 如果你需要你再学习，时间待定。SSH 的时间更待定。

三、我的反驳

主要在 Java 基础编程掌握程度：熟练。提到必须非常熟悉以下几个包，这几个包真有必要很熟练？javax.mail 我就从来没有用过，去学只会浪费时间。java.lang 我也有 N 个类根本不清楚，去了解那多干嘛，吃饱了没事干。其他的包也一样，去熟练掌握干嘛，时间没地方花是不是，那你还不如多出去散散步。中国的砖家太多，看来以上的基本要求，又不知是哪个砖家的杰作。

浩为的观点是学以致用，掌握基本原理后，在实践中学习，而不是一开始就背代码。很多人学习时先记住这个类怎么用，那个类又怎么用，越学越糊涂，因为要记的东西实在太多了。就如“**编程核心是代码，还是思想**”那篇所说，只讲代码，好比是告诉你一堆词组，这些词组是你从来就没有见过的，你能记住吗？而思想结合代码，很快就能了解，并记住了。

何况，代码在一个系统中到底占多大比重，我想大多数高手都知道。编程的核心是思想，语言只是思想的一种实现而已；当你有了思想，即使不懂某种语言，但你很快就能用其实现功能。在这思想的指导下，从没接触过 PHP 的陈聚雄十天就能参与项目开发，详情见 howwe.net。难道是这帮高手怕后来者很快就掌握了思想，一下就超过他们而使的一个诡计？不过要使这诡计也很正常，毕竟这帮高人是通过自己的长期摸索才成长起来，花费了许多时间，现在大多年老力衰。如果真把实质告诉新人，那这帮新人因为接收能力强，更是年轻力壮精力足，那他们很容易就能赶上高手。呵呵，那这帮高手不就岌岌可危咯。

代码只是一个过程，编程是一种思想，项目开发更是一个系统工程。对程序员来说，我想绝大多数绝不会只期望自己写一辈子代码，而是希望编码几年后就不再编码或少编码，成长为更高一级的 IT 人士。所以在编程时就得培养自己的整体思想，在学习时更应该去学习整体思想。

从职业发展来说，大多程序员先是只会写一点代码，随着经历的积累，慢慢掌握系统工程的思想，然后能写一些简单的文章，再做一些需求分析，再系统设计……也就是说，程序员最后都会掌握思想，既然如此，何不在一开始就在这种思想的指导下学习呢？

由于经常在 QQ 群里和群友们聊天，有人提出“技术 + 人生 = 数据库开发这点事”，我没反对，但我觉得更应该改为“技术 + 引导 = 数据库开发这点事”，因为引导包含人生、职业等多个方面。（注：这段话是在书名为《数据库开发这点事》时写的，书名修改后这话没改）

四、浩为对学习 Java 的建议

1、面向对象的知识：Java 是面向对象的开发语言，掌握面向对象的思想对学习 Java 很重要。面向对象的概念：对象、类、封装、多态、继承、抽象类、接口。了解概念后，还需要理解这些概念的关联，如类和对象有什么区别，类是如何封装？

2、Java 语法：其实也就是数据类型、控制类型、数据结构三种，再花点时间熟悉 Java 编程规范就可以了。

3、JSP 和 HTML：中国绝大多数公司，做 Java 的程序员都少不了和 JSP 及 HTML 打交道。

因此，要学习 JSP 和 HTML，知道 JSP 几大内置对象，如 Session, Request, Reponse, 以及常用的 JSP 标签，如 include, userBean 等。尽管一些工具能生成 HTML 代码，但还是要熟悉基本的 html 语法。如果熟悉 JS 和 CSS，那制作的页面会更友好。

4、WebServer: 熟悉了以上三点或第 3 点，就可以做页面了，要看到效果必须将页面放在 WebServer 上，如: Tomcat 或 Resin 等。

5、开发工具: 目前 Java 开发工具比较流行的有 Eclipse、Netbeans、Jbuilder、IDEA, HTML 开发工具有 Dreamweaver 等。

以上 5 点在书中都有提及，并对该书所用软件进行了整合，使用者简单配置就能搭建好开发环境，迅速进入学习阶段。在学习过程中，掌握本书提到的知识点，并抓住这些知识点的联系，为自己构建一个知识模型，就能起到纲举目张的作用，为你以后的发展打下坚实的基础。

为啥要对软件进行整合，因为整合后可以大大减少学习的时间及难度，就如前面砖家提到的“Tomcat 四天你可以掌握安装，配置”，可你用得着去浪费四天吗，使用整合软件，根本不用安装 JDK/JRE、Tomcat、Eclipse，简单运行运行脚本，一切都搞定，性能好的计算机，4 分钟都不需要，就能跑出一个 Jsp 页面。这对于初学者，第一个拦路虎不费吹灰之力就被搞定了。

对于编程来说，在掌握基本原理的情况下，很多细节的东西根本没有必要去了解，就如操作系统，有几人真正知道其原理，但是很多人能用。再去看看前面的一篇“**小语句解决大难题，IT 需要简单化**”，提到“就拿我自己来说，很少用 VBA，还是 99 年才用过半年的 VB，但我知道用 VBA 能解决问题，于是去 Google 资料，绝大多数人用的是 vlookup。7000 条数据，如果用 vlookup，需要很长时间，所以我就查使用 SQL 的方式。”

更提到以下内容：

其实用什么语言来编程，不是重要的。语言只是思维的体现，你解决问题的一种方式而已。所以我们更需要的是培养自己的思维，培养自己解决问题的能力。

拿以上的 VBA 来说，除了这一句：Sql = "select b1. [_COL0], b2. PARTY_ID, b1. [_COL1] from [nbs\$] as b1, [custom\$] as b2 where b1. [_COL1]=b2. PNAME" 之外，其他都是固定的，你也不用去了解为啥要这样写。就如使用操作系统一样，你只要会用，而根本不需要明白操作系统为啥能那样用。呵呵，你可以问问自己，你除了能使用之外，还了解多少操作系统的知识。道理一样，编程时，我们很多时候只要知道用什么能解决问题即可。

也许就会有砖家跳出来，指着我的鼻子说：“喻桃阳，你这小子，你这是误人子弟，这样教编程，你以为别人都和你一样是天才呀？”尽管嘴里说“天才”，心里却是“怪物”或“异端”。其实他更想说的是：“喻桃阳，你这小子，你还让不让我们活呀，你这样教编程，他们快速成长起来了，我们怎么办？”

编程更需要指导，回想自己的学习历程，走过不少弯路，不过还不是很多。因为我当初在图书馆看了不少有关系统设计的文章，明白代码在系统设计中所占比重不大，尽管也傻乎乎的抄过不少代码（记得是 96-97 年，那时没人指点，自己摸索），但在抄代码时更注重整体思想。所以我建议初学者也可以在整体思想的指导下去学习，这样就能快速成长起来。

比如对 Java 基础的学习，在掌握三大基础的情况下，再去看看 JDK 的 API 文档，搞清每个常用包的作用（即在你的脑海中建立一个模型，可用 FreeMind 自己做一个常用包的归纳图）就可以，用不着在每个类上花时间。学以致用，当你要用时，再去详细看一下 API 或 Google 一下，很快就掌握并能用了。

第 11 章 JSP+JavaBean

纯 JSP 页面其实就是在 HTML 代码里面写 Java 代码来完成逻辑。但这种方式存在不少问题：

1. Java 代码混杂在 HTML 中，显得混乱不堪，可读性非常差。一个 JSP 文件有时候会变成几十 K，甚至上百 K。要找一段逻辑，经常无法定位。
2. 编写代码时非常困惑，不知道代码到底应该写在哪里，也不知道别人是不是已经曾经实现过类似的功能，到哪里去引用。
3. 当需求发生变化后，修改很麻烦。
4. 逻辑处理程序需要自己来维护生命周期，对于类似数据库事务、日志等众多模块无法统一支持。

JavaBean 的出现，将页面上的 Java 代码抽取出来，只让页面保留尽量少的 Java 代码。即 JSP 只是表示层，JavaBean 为业务层。

JavaBean 常用来封装事务逻辑、数据库操作等，很好地实现业务逻辑、前台程序(如 JSP 文件)的分离，使得系统具有更好的健壮性和灵活性，可以实现代码的重复利用，并提高了程序的易维护性。

我们先以 Hello World 为例，示范一个简单的例子；然后结合电话外呼系统的页面，看看实际中的页面是怎么样子的。

11.1 JavaBean 版 Hello World

在项目 howwe 的 src\db 目录上单击右键，然后选择 **New => File**，为文件名称为 d11101vo.java，然后按 Finish。

在文件中输入如下代码并保存：

```
package db;

public class D11101vo {
    String cnt = null; //对应表字段cnt

    public String getCnt() {
        return cnt;
    }

    public void setCnt(String cnt) {
        this.cnt = cnt;
    }
}
```

此文件在用于在数据操作中传递数据。

在项目 howwe 的 src\db 目录上单击右键, 然后选择 **New => File**, 为文件名称为 d11101.java, 然后按 Finish。

在文件中输入如下代码并保存:

```
package db;

import java.util.ArrayList;
import org.apache.log4j.Logger;

public class D11101 {
    private static Logger log = Logger.getLogger(D11101.class);
    String sSql="";

    public D11101() {

    }

    @SuppressWarnings("unchecked")
    public ArrayList recRead() {
        ArrayList alResult = new ArrayList();
        //定义变量
        java.sql.Connection conn = null; //数据库连接
        java.sql.Statement st = null; //执行语句对象
        java.sql.ResultSet rs=null; //记录集
        String sSql=null;
        try {
            //创建数据库连接
            Class.forName("org.hsqldb.jdbcDriver"); //加载数据库驱动
            conn=java.sql.DriverManager.getConnection("jdbc:hsqldb:hsqldb://127.0.0.1:19132/webapp", "sa", "sa1"); //数据库连接, 采用SERVER访问模式
            st=conn.createStatement(); //创建执行语句对象
            //查询记录
            sSql="SELECT cnt FROM test";
            rs = st.executeQuery(sSql);
            while (rs.next()) {
                D11101vo fd = new D11101vo();
                fd.setCnt(rs.getString("cnt"));
                alResult.add(fd);
            }

            //释放资源
            rs.close(); //关闭记录集
            st.close(); //关闭执行语句对象
            conn.close(); //关闭数据库连接
        } catch (Exception e) {
            log.error(e.getMessage());
        }
    }
}
```

```
    } catch (Exception ex) {  
        log.error("", ex);  
    }  
    return alResult;  
}  
}
```

此文件在用于操作数据，方法recRead读取记录并返回ArrayList对象。

在项目 howwe 的 webroot 目录上单击右键，然后选择 **New => File**，为文件名称为 d11101r.jsp，然后按 Finish。

在文件中输入如下代码并保存：

```
<%@ page contentType="text/html;charset=gbk"%>  
<%@ page import="db.*"%>  
<%@ page import="java.util.ArrayList"%>  
<%@ page import="java.util.Iterator"%>  
  
<%  
try {  
    //引用Bean  
    D11101 recDao = new D11101();  
  
    //调用方法：查询记录  
    ArrayList recList = recDao.recRead();  
%>  
  
<HTML>  
<BODY>  
<%  
    if (recList.size() > 0) {  
        Iterator itr = recList.iterator();  
        while (itr.hasNext()) {  
            D11101vo rd = (D11101vo)itr.next();  
%>  
            数据库内容为: <%=rd.getCnt() %><BR>  
%>  
        }  
    }  
} catch (Exception e) {  
    e.printStackTrace();    //显示数据操作出错原因  
}  
%>  
</BODY>  
</HTML>
```



(图 11-1 浏览页面-读数据库)

原理：用户通过浏览器来发送对网页的请求，服务器收到请求后，在服务器端查找对应的网页，如果是首次请求，如果是 JSP 页面，先要生成 Servlet（第二次就不用了），然后通过 Servlet 引擎来执行 Servlet，把调用 JavaBean 的结果嵌入到页面中返回给用户的浏览器。

JSP 页面通过调用 JavaBean（即数据操作类）中的数据操作方法，返回数据对象，然后根据 VO 来操作具体的数据，从而实现对数据的处理。VO，Value Object 的简称，值对象，通常用于业务层之间的数据传递，简单说就是一种数据结构，里面可以放数据。

如图 11-1 所示的页面 d11101r.jsp 通过调用 D11101 中的 recRead 方法，返回的数据通过 D11101vo 中的 get 方法来读取数据，从而实现数据访问。

形象一点说，D11101vo 就是一个果盘，D11101 中 recRead 的作用是把一堆放果实的果盘从仓库（数据库）里搬到果架 d11101r.jsp，再由果架把果实从果盘里取出来放在果架上。

而单纯的 JSP 页面呢，就相当于直接从仓库取果实摆在果架上。

11.2 运营商管理页面

HwCall 浩为电话外呼系统，详情请参考“第 14 章 电话外呼系统”，下面我们介绍其中的页面：运营商管理->查询。

单击 http://127.0.0.1 页面上的链接：运营商管理（或直接在浏览器地址栏输入 URL：http://127.0.0.1/demo/d111020pDec1.jsp），界面如图 11-2 所示

运营商管理->查询					
业务大类: <input type="text"/>					
运营商种类: <input type="text" value="--请选择运营商--"/>					
业务别名: <input type="text"/>					
<input type="button" value="查询"/> <input type="button" value="重填"/> <input type="button" value="添加"/>					
ID	业务大类	运营商种类	业务别名	时间	操作
3	联通气象	中国联通	lt	2009-01-14 16:55:48	编辑
2	电信气象	中国电信	dx	2008-11-18 22:13:58	编辑
1	移动气象	中国移动	yd	2008-07-23 07:01:00	编辑

(图 11-2 运营商管理-查询)

该页面对应的文件为 d111020pDec1.jsp（根据 smscOpDec1.jsp 简单修改而来），调用的数据操作类为 SmsDaocOpDec1，对应的 VO 数据结构为 SmsDataOpDec1，具体代码请自己看范例。

需简单说明的代码如下（smcOpDecl.jsp 中多的语句）：

```
<jsp:include page="common/admin.jsp">
  <jsp:param name="Promise" value=",801,"/>
</jsp:include>
<%@include file="common/validate.jsp"%>
```

前三句用于鉴权，后一句引入基本的对象及设置。

至此，Jsp+JavaBean 的介绍基本结束，剩下的就是如何在具体程序中去应用和发挥。

第 12 章 数据结构

在计算机科学中，数据结构（data structure）是计算机中存储、组织数据的方式。数据结构是指相互之间存在一种或多种特定关系的数据元素的集合。

通常情况下，精心选择的数据结构可以带来最优效率的算法。一般而言，数据结构的选择首先会从抽象数据类型的选择开始。一个设计良好的数据结构，应该在尽可能使用较少的时间与空间资源的前提下，为各种临界状态下的运行提供支持。数据结构可通过编程语言所提供的数据类型、引用及其他操作加以实现。

不同种类的数据结构适合于不同种类的应用，而部分甚至专门用于特定的作业任务。例如，当计算机网络依赖于路由表运作时，B 树就可用于数据的封装。

在程序设计中，选择适当的数据结构是一个主要的考虑因素。许多大型系统的构造经验表明，封装的困难程度与最终成果的质量与表现，都取决于是否选择了最优的数据结构。在许多时候，确定了数据结构后便能很容易地得到算法。而有些时候，方向则会颠倒过来：例如当某个关键作业需要特定数据结构下的算法时，会反过来确定其所使用的数据结构。然而，不管是哪种情况，数据结构的选择都是至关重要的。

系统构造的关键因素是数据结构而非算法，导致了多种形式化的设计方法与编程语言的出现。绝大多数的语言都带有某种程度上的模块化思想，通过将数据结构的具体实现封装隐藏在接口里面的方法，来让不同的应用程序能够安全地重用这些数据结构。C++、Java、Python 等面向对象的程序设计语言可使用类来完成这一功能。

因为数据结构的重要性毋庸置疑，现代编程语言及其运行环境在标准库中都包含了多种的数据结构，例如 C++ 标准模板库中的容器、Java 集合框架以及微软的 .NET Framework。

大多数数据结构都由数列、记录、可辨识联合、引用等基本类型构成。举例而言，可空引用（null able reference，一种可被置空的引用）是引用与可辨识联合的结合体，而最简单的链式结构链表则是由记录与可空引用构成。

数据结构意味着接口或封装：一个数据结构可被视为两个函数之间的接口，或者是由多个数据类型联合组成的存储内容的访问方法的封装。

12.1 基本概念

Robert L. Kruse 在《数据结构与程序设计》一书中, 将一个数据结构的设计过程分成抽象层、数据结构层和实现层。其中, 抽象层是指抽象数据类型层, 它讨论数据的逻辑结构及其运算, 数据结构层和实现层讨论一个数据结构的表示和在计算机内的存储细节以及运算的实现。

一个数据结构是由数据元素依据某种逻辑联系(或关系)组织起来的。对数据元素间逻辑关系的描述称为数据的逻辑结构; 数据必须在计算机内存储, 数据的存储结构是数据结构的实现形式, 是其在计算机内的表示; 此外讨论一个数据结构必须同时讨论在该类数据上执行的运算才有意义。

在许多类型的程序的设计中, 数据结构的选择是一个基本的设计考虑因素。许多大型系统的构造经验表明, 系统实现的困难程度和系统构造的质量都严重的依赖于是否选择了最优的数据结构。许多时候, 确定了数据结构后, 算法就容易得到了。有些时候事情也会反过来, 我们根据特定算法来选择数据结构与之适应。不论哪种情况, 选择合适的数据结构都是非常重要的。

数据(Data)是信息的载体, 它能够被计算机识别、存储和加工处理。它是计算机程序加工的原料, 应用程序处理各种各样的数据。计算机科学中, 所谓数据就是计算机加工处理的对象, 它可以是数值数据, 也可以是非数值数据。数值数据是一些整数、实数或复数, 主要用于工程计算、科学计算和商务处理等; 非数值数据包括字符、文字、图形、图像、语音等。

数据元素(Data Element)是数据的基本单位。在不同的条件下, 数据元素又可称为元素、结点、顶点、记录等。例如, 学生信息检索系统中学生信息表中的一个记录、八皇后问题中状态树的一个状态、教学计划编排问题中的一个顶点等, 都被称为一个数据元素。

有时, 一个数据元素可由若干个数据项(Data Item)组成, 例如, 学籍管理系统中学生信息表的每一个数据元素就是一个学生记录。它包括学生的学号、姓名、性别、籍贯、出生年月、成绩等数据项。这些数据项可以分为两种: 一种叫做初等项, 如学生的性别、籍贯等, 这些数据项是在数据处理时不能再分割的最小单位; 另一种叫做组合项, 如学生的成绩, 它可以再划分为数学、物理、化学等更小的项。通常, 在解决实际问题时, 把每个学生记录当作一个基本单位进行访问和处理。

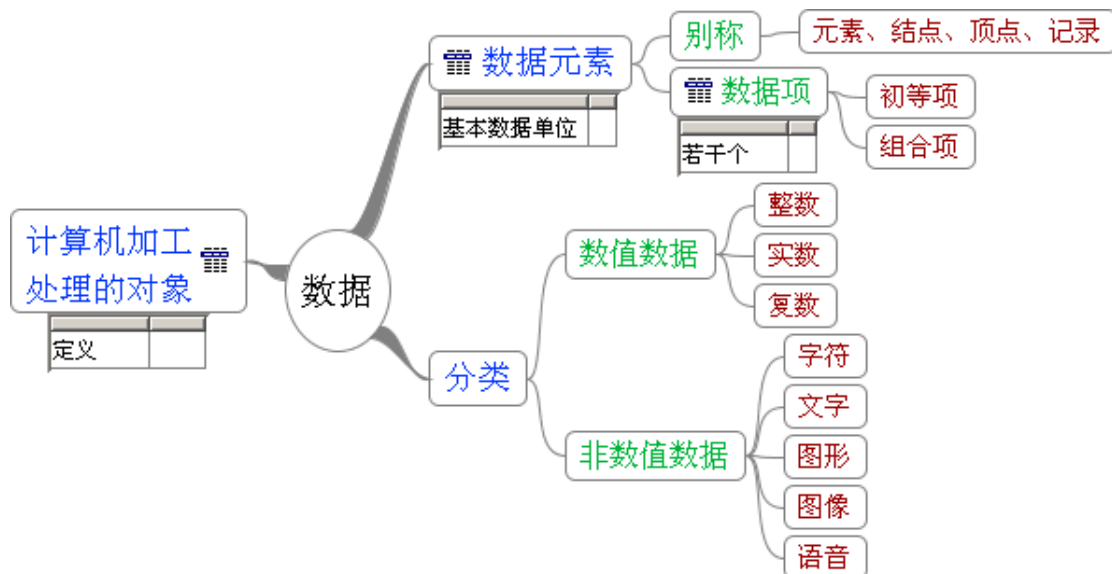
数据的概念如图 12-1 所示, 显示了数据与数据元素的关系。数据范例如图 12-2 所示, 示范学生信息表的构成。

数据对象(Data Object)或数据元素类(Data Element Class)是具有相同性质的数据元素的集合。在某个具体问题中, 数据元素都具有相同的性质(元素值不一定相等), 属于同一数据对象(数据元素类), 数据元素是数据元素类的一个实例。例如, 在交通咨询系统的交通网中, 所有的顶点是一个数据元素类, 顶点 A 和顶点 B 各自代表一个城市, 是该数据元素类中的两个实例, 其数据元素的值分别为 A 和 B。

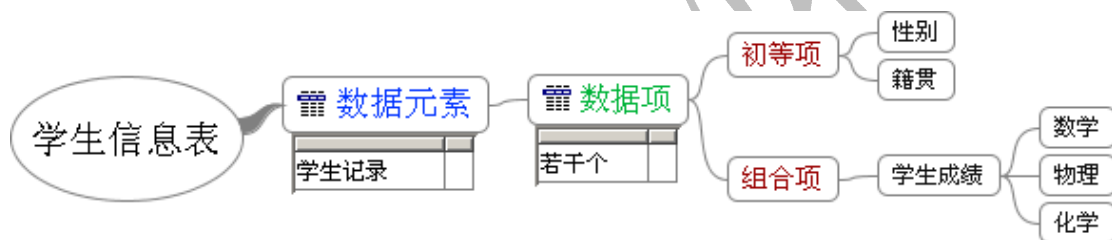
数据结构(Data Structure)是指互相之间存在着一种或多种关系的数据元素的集合。在任何问题中, 数据元素之间都不会是孤立的, 在它们之间都存在着这样或那样的关系, 这种数据元素之间的关系称为结构。根据数据元素间关系的不同特性, 通常有下列四类基本的结构:

(1)集合结构。在集合结构中, 数据元素间的关系是“属于同一个集合”。集合是元素关系极为松散的一种结构。

- (2)线性结构。该结构的数据元素之间存在着一对一的关系。
- (3)树型结构。该结构的数据元素之间存在着一对多的关系。
- (4)图形结构。该结构的数据元素之间存在着多对多的关系，图形结构也称作网状结构。



(图 12-1 数据概念)



(图 12-2 数据范例)

12.2 基础数据结构

数据结构包含以下知识：

几种基本的线性结构：线性表、栈、队列、串等，几种非线性结构：多维数组、广义表、树、图、矩阵、哈希表、堆、优先队列、空间数据结构等，常用的数据处理技术：插入、删除、排序、查找等，文件的存储结构和组织方法等。

但在实际编程工作中，大部分数据结构都不会用到，而且也许永远都不会用到。尽管如此，深入地理解基本数据结构的概念和实现细节，仍然是每一个程序员的任务。这不仅是因为，掌握这些知识，将有利于更加正确和灵活地应用它们。

下面介绍三种最常用的数据结构：字符串、数组和 Hash 表，比较它们在不同语言中的实现思路，及它们的使用技巧。

1、字符串

严格地说，字符串(string)甚至不能算作一种单独的数据结构，至少在 C 语言中，它仅仅是某种特定类型的数组而已。但是，字符串在实际使用中很重要，而且在不同语言中的实现差异不少。

最经典的字符串实现, 是 C 语言中的零终结 (null-terminated) 字符串, 实质上是一个字符数组, 它依次存放字符串中的每个字符, 最后以零字符 ('', 表示为常量 null) 作为结束, 对字符串的所有操作, 都是通过对字符数组的操作来完成。因此, C 语言字符串占据的空间比它实际的长度要多 1 个单元。在实际应用中, 它常以数组或字符指针的形式被定义, 如下例:

```
char[] message = "this is a message";  
char* pmessage = "an other message";
```

针对 C 语言字符串的缺陷, 新的语言进行了相应的改进。

C++ 语言在标准库中提供了一个基础字符串的实现: `std::basic_string`。它封装了大量常见的操作, 例如取长度、比较、插入、拼接、查找、替换等等, 并且能够自动管理内存。

在 Delphi 中, 字符串仍然是一种基本类型, 而不是类。它的实现方式也是字符数组, 但在数组的头部增加了两个 32 位整数存储空间, 分别用于存放字符串的长度和引用计数。通过前者可以方便地获得字符串的长度, 而不需要进行无谓的遍历操作。后者实现了 COW (Copy on Write) 技术, 这种技术的效果是: 当字符串被复制时, 并不会复制其内容, 而只是建立一个新的指针, 指向原有的字符串, 并在引用计数上加一。当字符串被删除时, 引用计数减一, 当引用计数为 0 时, 字符串的内存将被释放。只有当对字符串进行写入操作时, 才会建立一个新的字符串并复制内容。这些工作是由编译器自动完成。

Java 和 C# 中的字符串, 是一个封装了常见操作的类, 这一点和 C++ 类似。一个特殊之处 (往往导致经典的性能问题) 是, 无论是在 Java 还是在 C# 中, String 类都是不变 (immutable) 的。也就是说, String 的内容不能够被改变, 如果代码试图改变一个 String 对象的内容, 实际的结果是建立了一个新的 String 对象, 并抛弃旧的对象。如下例:

```
String s = "";  
for (int i = 0; i < 10000; i++) {  
    s += i + ", ";  
}
```

结果是建立并抛弃了 10000 个 String 对象, 这在性能上的开销是惊人的。为了避免这种情况, 应该使用 StringBuilder 对象, 它可以改变其内容。(C# 一直使用 StringBuilder。Java 从 1.5 开始引入 StringBuilder 以部分替代 StringBuffer, 它们的主要区别在于线程安全性。) 如下例:

```
StringBuilder sb = new StringBuilder();  
for (int i = 0; i < 10000; i++) {  
    sb.append(i + ", ");  
}
```

2、数组

从抽象数据类型的意义上来说, 一维数组 (array) 的定义是: 具有相同数据类型的若干个元素的有限序列。

在 C 语言中, 数组意味着一块连续的内存空间, 按顺序存放着若干个相同数据类型的元素。可以通过下标来访问数组中的元素。如下例:

```
int a[10]; // 定义一个 int 型的数组  
for (int i = 0; i < 10; i++) {  
    a[i] = i; // 赋值  
}
```

在 C 语言中, 数组名实际上是一个指针 (指向该数组的第一个元素), 因此所有通过数组下标完成的操作, 都可以通过指针来完成。通过指针来访问数组, 效率上比数组下标要高, 而且更加灵活, 例如, 指针可以进行偏移量的运算, 甚至可以进行绝对地址的存取。

C 语言中的数组没有越界检查, 这意味着, 程序员可以访问数组最后一个元素以后的地址, 或者第一个元素之前的地址 (例如, `a[-1]`、`a[-2]` 这种形式是合法的)。在某些情况下, 这是一种有用的技巧, 但大多数情况下是一场灾难。C 语言的数组也不支持自动增长, 如果数组的长度发生了变化, 程序员必须手动处理所有关于申请和释放内存的工作。

C++ 提供了 C 风格的数组, 同样不支持越界检查和自动增长。但是, C++ 建议, 应该尽量使用 STL 中的容器作为替代品, 一般是 `vector`。`Vector` 基于面向对象和模板技术, 构建了一个强大而复杂的类, 实现了如下特性: 高效率的自动内存管理; 按任何顺序访问、插入和删除元素; 越界检查, 但同时也提供了不进行检查的访问方式, 以照顾性能上的考虑; 基于运算符重载技术的运算符支持; 基于迭代器的漫游机制; 与数据类型无关的算法支持; 等等。相对于 C 风格的数组, `vector` 是一种更高抽象层次上的序列概念。它对大量常用的功能进行了封装 (例如, 对内存的直接操作), 同时又尽可能地照顾了效率和可移植性 (例如, 在自动扩充时通过缓存机制来提高效率)。这也正是 C++ 语言对 C 语言进行改进时的指导思想。

Delphi 也支持 C 风格的数组, 但提供了越界检查, 还提供了一种动态数组 (`Dynamic Array`), 可以在运行时通过 `SetLength` 函数动态地改变它的大小。事实上, `SetLength` 函数就是对内存管理操作的一种封装。类似于 C++ 中的 `vector`, Delphi 也提供了两个可以自动增长的容器: `TList` 和 `TObjectList`, 前者用于存放无类型的指针, 后者用于存放对象。由于 Delphi 不支持模板机制, 所以 `TList` 不会自动释放指针所指向的内存, 它只会维护指针自身占用的内存 (`TObjectList` 能够在销毁时自动释放元素所占用的空间, 如果它的 `OwnsObjects` 属性被设置为 `True` 的话)。

Java 也支持加上了越界检查的 C 风格数组, 但它提供的类似容器更为引人注目。Java 将序列 (`List`) 作为一个单独的接口提取出来, 并提供了两个实现: `ArrayList` 和 `LinkedList`。从名字就可以看出来, 前者是通过数组来实现的, 后者则通过链表。

由于都实现了 `List` 接口, 二者可以支持同样的基本操作方式, 不同的是, `ArrayList` 在频繁进行随机访问时有效率优势, 而 `LinkedList` 在频繁进行插入和删除操作时效率较优。

3、Hash 表

Hash 表的实现思路如下: 通过某种算法, 在键-值对的存储地址和键-值对中的 `key` 之间, 建立一种映射, 使得每一个 `key`, 都有一个确定的存储地址与之对应。这种算法被封装在 Hash 函数中。在查找时, 通过 Hash 函数, 算出和 `key` 对应的存储地址, 从而找到相应的键-值对。相对于通过遍历整个键-值对列表来进行查找, Hash 表的查找效率要高得多, 理想的情况下算法复杂度仅为 $O(1)$ (遍历查找的复杂度为 $O(n)$)。

但是, 由于通常情况下 `key` 的集合比键-值对存储地址的集合要大得多, 所以有可能把不同的 `key` 映射到同一个存储地址上。这种情况称为冲突 (`collision`)。一个好的 Hash 函数应该尽可能地把 `key` 映射到均匀的地址空间中, 以减少冲突。Hash 表的实现也应该提供解决冲突的方案。

Hash 表是一种相对复杂得多的数据结构, 从底层完整地实现一个 Hash 表, 也许超出了对一个普通程序员的要求。但是, 由于它是如此重要, 了解 Hash 表的概念和掌握使用它的接口, 仍然是一项必不可少的技能。

C 语言中没有提供现成的 Hash 表, 但是 C++ 提供了优秀的 Hash 表实现容器 `hash_map`。

象 STL 中的其他容器一样，hash_map 支持任何数据类型，支持内存自动管理，能够自动增长。特别地，hash_map 通过模板机制，实现了和 hash 函数的剥离，也就是说，程序员可以定义自己的 hash 函数，交给 hash_map 去进行相应的工作。如下例：

```
hash_map <string, int> hml; // 使用默认的 Hash<string>函数
hash_map <string, int, hfct> hml; // 使用自定义的 hfct() 作为 hash 函数
hash_map <string, int, hfct, eql> hml; // 使用自定义的 hfct() 作为 hash 函数，
并且使用自定义的 eql() 函数比较对象是否相等
```

Java 定义了 Map 接口，抽象了关于 Map 的各种操作。在实现了 Map 接口的类中，HashMap 是 Hash 表，接受任何类型的对象作为键-值对的元素，支持快速的查找。如下例：

```
HashMap hm = new HashMap();
hm.put("akey", "this is a word"); // 使用两个字符串作为键-值对
String str = (String) hm.get("akey");
System.out.println(str);
```

HashMap 和 hash 函数也是剥离的，但使用了另一种思路。在 Java 中，根类型 Object 定义了 hashCode() 和 equals() 方法，由于任何类型的对象都派生自 Object，所以它们都自动继承了这两个方法。用户自定义的类应该重载这两个方法，以实现自己的 hash 函数和比较函数。如果这两个函数没有被重载，Java 会使用 Object 的 hashCode() 和 equals() 方法，它们的默认实现分别是返回对象的地址，以及比较两个对象的地址是否相等。

第 13 章 JSP+JavaBean+Servlet

JSP+JavaBean 两层结构很容易理解，开发上手也很快。

JSP+JavaBean+Servlet 三层结构的实质只是多了一个 Controller: Servlet 来分发及处理来自客户端浏览器的请求。现实中，许多人使用 servlet 处理业务逻辑。故在本章不详细讲述，只介绍基本原理。

通过 web.xml 配置文件可以找到用户请求和特定的 Servlet 的对应关系，每个 Servlet 都有一个特定的 Servlet 对象与之对应，所以说处理用户请求的就是一个继承自 HttpServlet 的 Servlet 对象。

13.1 Servlet 版 Hello World

在项目 howwe 的 src\db 目录上单击右键，然后选择 **New => File**，为文件名称为 d11302Hello.java，然后按 Finish。

在文件中输入如下代码并保存：

为了缩减篇幅，代码略，请自己查看 howwe-ok 项目 db/d11302Hello.java，文件具体路径：D:\work\hocat\hocas-ok\src\db\d11302Hello.java。

解析：doPost 方法处理具体的请求。

打开项目 howwe，目录 webroot\WEB-INF 下面有一个文件 web.xml，添加如下代码并保存（放在<welcome-file-list>之前）：

```
<servlet>
    <servlet-name>Hello</servlet-name>
    <servlet-class>db.D11302Hello</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>Hello</servlet-name>
    <url-pattern>/hello</url-pattern>
</servlet-mapping>
```



(图 13-1 Servlet 版 Hello World)

13.2 应用系统中的 Servlet

打开项目 howwe-ok，目录 webroot\WEB-INF 下面有一个文件 web.xml，其中有如下代码：

```
<servlet>
    <servlet-name>InitApp</servlet-name>
    <servlet-class>net.howwe.core.InitApp</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>InitApp</servlet-name>
    <url-pattern>/key</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>InitApp</servlet-name>
    <url-pattern>/db</url-pattern>
</servlet-mapping>
```

第一部分（<servlet>与</servlet>），主要用来配置 Servlet 与具体的 Servlet 对象之间的关联，第二部分（<servlet-mapping>与</servlet-mapping>），主要用来配置请求由哪个 Servlet 处理，即请求与具体 Servlet 处理对象关联起来。

例，在浏览器地址栏输入 http://127.0.0.1/demo/key，界面如图 13-2 所示，它由 InitApp servlet 进行处理，通过 InitApp 就能找到相对应的 servlet 对象 net.howwe.core.InitApp，即

请求 /key->InitApp->net.howwe.core.InitApp, 这就是配置文件的意义所在。

注意:

- 1、在servlet之间加入load-on-startup, 可使servlet按顺序加载;
- 2、多个servlet-mapping可对应一个servlet, 但一个servlet-mapping不可对应多个servlet。

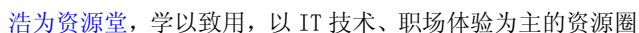


(图 13-2 key 请求)

相关代码如下, 具体请查看 net.howwe.core.InitApp 中的 doPost。

```
sRtn=request.getQueryString();
if (sRtn==null)sRtn="";else sRtn="?" +sRtn;
sRtn=request.getRequestURI()+sRtn;

if (sRtn.indexOf("/key")>0) {
    String key1=du.DecSc(request.getParameter("key1"), "");
    String key2=du.DecSc(request.getParameter("key2"), "");
    String key3=du.DecSc(request.getParameter("key3"), "");
    int step=du.DecInt(request.getParameter("step"), 0);
    if (step==1&&!key1.isEmpty()) {sRtn="授权范例! \n";}
    String sInfo=dw.doStr("select info from TSMSCINFO", 1);
    sRtn = "<html><head><title>howwe.net 版本授权</title>"
        + "<meta http-equiv=Content-Type content=text/html; charset=gbk>\n"
        + "<link rel=\"stylesheet\" href=\"img/css/cp.css\" "
        + "type=\"text/css\">\n"
        + "</head>"+sRtn+"<body leftmargin=2 topmargin=2 "
        + "background=\"img/bg.gif\">\n"
        + "<form action=\"?step=1\" method=\"post\" name=\"addform\">\n"
        + "<table width=\"100%\" align=\"center\" border=\"0\" "
        + "cellspacing=\"1\" cellpadding=\"4\" class=\"tableoutline\">\n"
        + "<tr id=\"cat\"><input type=hidden name=dact "
        + "value=\"web/common/calendar/jcalendar\">"
        + "<td class=tbtitle colspan=\"4\"><b>授权</b></td></tr>\n"
```



第 14 章 电话外呼系统

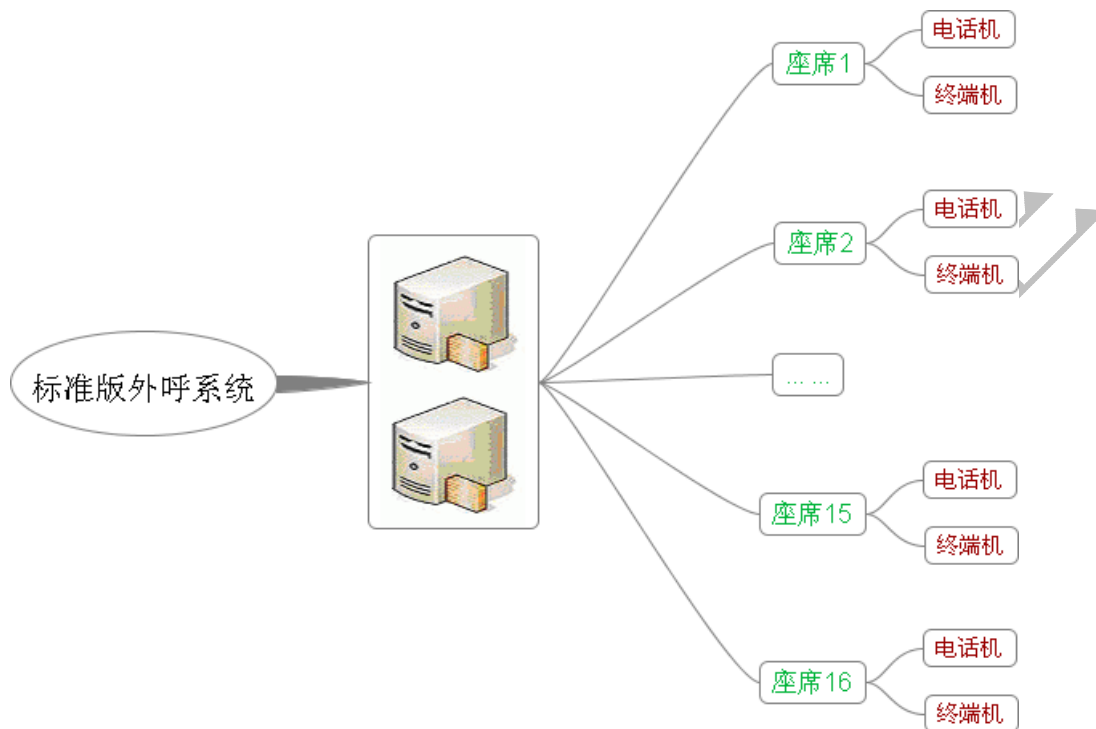
我们先看电话外呼系统的组成，然后介绍系统的安装及操作等。下两章介绍系统设计及编码。

14.1 硬件配置方案

方案中提供两种版本的硬件方案，具体配置可以根据业务量来选择。

标准版硬件由两个服务器和 16 或 32 套终端机组成。录音卡、语音管理系统放在一台服务器，外呼号码管理系统、终端管理系统放在一台服务器。

标准版系统的拓扑结构图如下：



（图 14-1 标准版外呼系统拓扑结构）

基础版硬件由一个服务器和 8 套终端机组成。录音卡、外呼号码管理系统、语音管理系统、终端管理系统放在一台服务器，最大可支持 16 套终端。

系统拓扑结构同图 14-1，只是服务器变成了一个。

一、录音部分

采用 MDR9500 电话录音系统，基本功能如下：

- 多种压缩率（0-不压缩<1/1>，1-ADPCM<1/2>，2-GSM<1/5>）可供选择。
- 当采用 GSM 压缩率时，40G 硬盘可存储 7000 小时的语音文件。
- 所录音文件为 WAV 格式，可在 WINDOWS 的媒体播放器中播放。
- 可连接多种通信线路，如电话外线、分机线、对讲机等。三种启动方式对应于特定线路可分别设定。
- 主画面显示系统录音状况以及系统使用的录音空间与剩余录音空间。空闲、录音、监听、不使用等图形对照其工作状态，使系统的录音作业获得即时掌控。
- 可以完整的检测到外拨电话号码与来电显示号码。
- 方向判别。可以判别出呼叫方向，如呼入还是拨出。
- 用户权限。系统至少可以设置 10 种权限，以保证不同行业的需要。

二、终端机

使用 3000-A 电脑共享器，可以使一个主机最大可带三十个显示器实现独立的操作。30 个用户同时外呼，互不干扰！



(图 14-2 终端机)

3000-A 是以网络为基础的独立终端机，它是一台不需 CPU、硬盘等常备硬件的电脑终端，允许多个用户共同使用一台主机的 CPU、内存、硬盘、驱动器等资源，操作与主机一样。终端通过快速的以太网 TCP / IP 协议来连接主机或服务器，一台 windows 2000/2003 服务器版主机最多可以支持 30 个终端，各用户同时独立安全地进行外呼。

14.2 软件部分

语音管理系统由录音卡厂家提供，终端管理软件由终端机厂家提供。

外呼号码管理系统采用浩为开发的 HwCall，可以实现外呼的无纸化办公及管理。

基本操作流程：

- 1、管理员通过“电话导入”将号码导入系统；
- 2、回访员根据“回访单”显示的号码拨打电话，打完后归类（类的配置在“业务配置”->“业务管理”中进行，其中序号为 11-20 规定为有效的业务、21-26 为无效的业务，可以随意调整；27 为预订，可调整或去掉；28-29 为固定的无效业务，不可调整）；
- 3、管理员对回访员的工作及号码进行管理；
- 4、回访单产生的有效用户，可以通过“电话回访”->“电话查询”，进行修改。其中预订也属于有效用户。

注：现已推出“浩为电话营销系统HwCall 3.0”，基于V08-PCI语音卡开发的电话营销系统HwCall 3.0，类似于呼叫中心，能解析用户按键、自动录音，监听，管理及播放录音。如果采用相应语音卡，就能为用户自动播放语音、转内线分机等，成为一个完整的呼叫中心或自动外呼系统。简介：<http://howwe.net/corp/150.html>。

14.3 安装准备

HwCall 电话外呼系统采用 Java 开发，在浏览器中运行。名为 howwe.rar 的压缩文件包含相关软件。安装前如没有安装 WinRar 的压缩软件，请先安装。

将 howwe.rar 解压到 D 盘根目录下，注意“运行必读.txt”（如没有此文件，则为“运行必读”，下同）必须在 d:\howwe 目录下，否则，此系统无法正常运行。

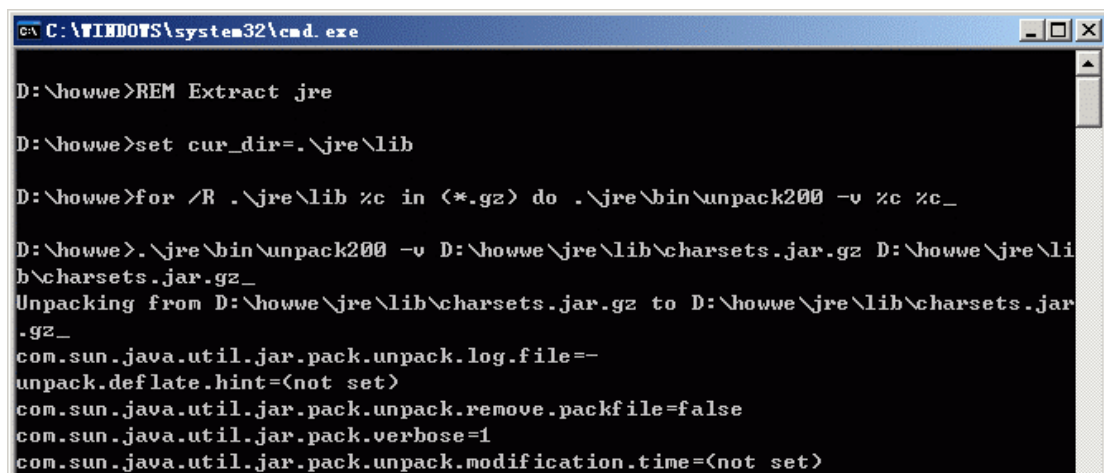
HwCall 自带 Java 运行环境 JRE，内存数据库 Hsql，采用 Resin 作为 Web 服务器。企业版可采用 mysql 或 mssql 数据库，同时采用 Hsql 作为缓存数据库。

下载地址：<http://howwe.net/corp/150.html>

14.4 系统安装

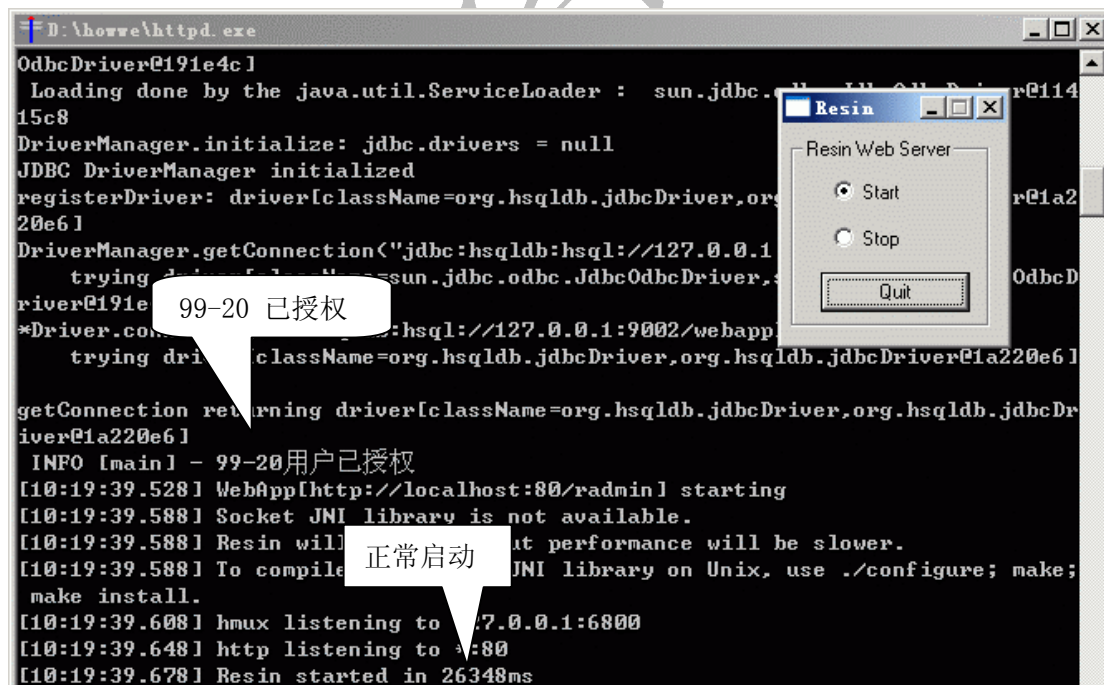
先运行（鼠标双击）D:\howwe\0 安装.bat

如图 14-3 所示，开始解压被压缩的运行程序，解压完成，此窗口将自动关闭，0 安装.bat 也被删除。



（图 14-3 解压运行程序）

待运行程序解压结束后，就可以运行系统，鼠标双击 D:\howwe\1 运行.bat，将启动 httpd.exe，此文件为 Web 服务器。正常运行后，其他终端才能通过浏览器 IE 或 Firefox 使用本系统。

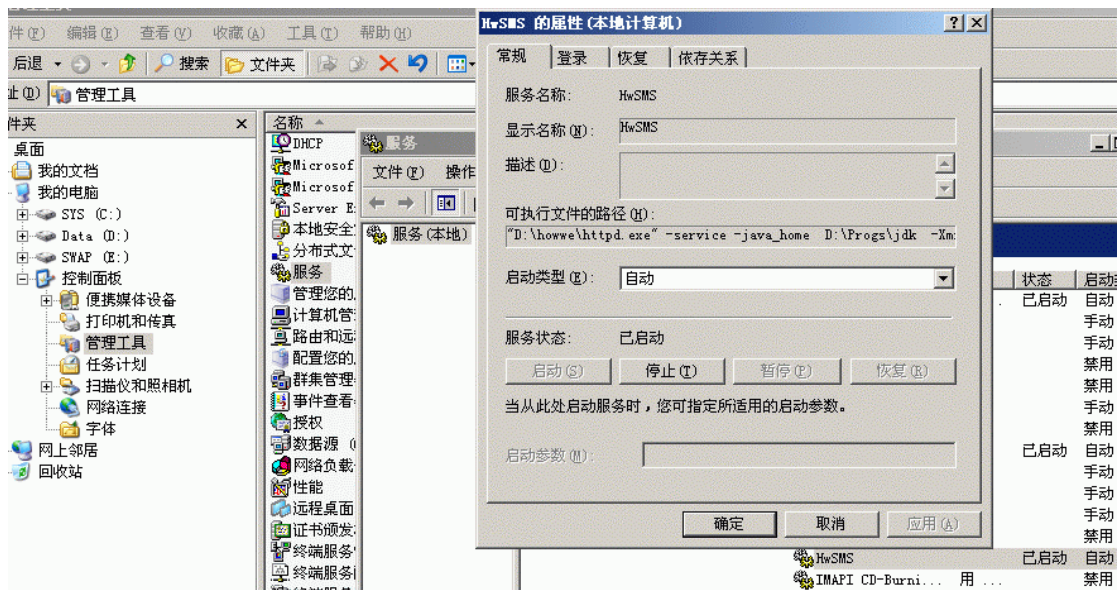


（图 14-4 系统运行中）

运行正常，将显示：Resin started in **** ms。注意，如要系统正常运行，图 14-4 所示的两个窗口都不能关闭。如要重启，可以直接关闭窗口后再运行。

如需此系统随着服务器的启动而自动加载，请运行 D:\howwe\bin\设置为自启动.bat。在这种情况下，如果要重启此系统，请选择：资源管理器—》控制面板—》管理工具，打开

“服务”，再选择名为“HwSMS”的服务，鼠标双击打开如图 14-5 所示的窗口，先停止，再启动即可。



(图 14-5 系统注册为服务后的重起)

运行 D:\howwe\2 浏览.bat，将启用浏览器查看网页 http://127.0.0.1。登陆时用户名 admin，密码为空。（密码在操作员模块修改）

注意：

- 1、授权版本，请勿修改电脑名称、安装目录下的文件（hwcass\data 除外）、调整网卡，否则版本将变成未授权。
- 2、硬件相关软件的安装请参照硬件厂家提供的安装说明书。
- 3、如果服务器重装，保持电脑名称不变，d:\howwe 下的文件不变，可不用重新授权。

14.5 用户种类及功能模块

用户分为两类：管理员、回访员。

回访员一般只分配“回访单”这个模块，登陆后直接进入此模块。注意，回访员登陆时可以选择待回访的用户种类：中国移动、中国电信、中国联通之一。

管理员能使用全部模块。

一、功能列表

1、业务配置

- 1.1、运营商管理：配置运营商大类（如联通气象、移动气象）
- 1.2、业务管理：配置具体业务种类（如完成、拒绝）
- 1.3、地区配置：配置地区

2、电话管理

- 2.1、电话导入：管理员将待回访或剔除的号码导入系统
- 2.2、电话导出：查询或导出指定号码
- 2.3、话务统计：对回访员的工作进行统计，并可导出相应的回访号码

3、电话回访

- 3.1、回访单：回访员对号码进行回访并归类
- 3.2、回访统计：回访员对自己业务的统计
- 3.3、电话查询：回访员查询指定号码，并可修改几种号码的归类

4、系统维护

- 4.1、操作员：管理可使用该系统的用户

二、数据管理

提供如下界面对数据进行特别处理，建议少用。



(图 14-6 数据管理)

操作说明：用户以admin成功登陆后，修改浏览器地址栏的地址如图 14-6 所示，<http://127.0.0.1/web/db>，即可出现如图 14-6 所示的界面。

- 1、“重新回访”只能针对“拒绝”、“为接通”两类号码。
- 2、“删除”将全部删除选定的用户。

注意：未授权版本不能使用此功能。

14.6 用户操作

在浏览器地址栏输入<http://127.0.0.1/demo>，如图 14-7 所示，点击 **登陆** 按钮，登陆后左边菜单展开后如图 14-8 所示。



(图 14-7 登陆)



(图 14-8 HwCall 菜单)

单击菜单：运营商管理，界面如图 14-9 所示

运营商管理->查询					
业务大类:					
运营商种类:		--请选择运营商--			
业务别名:					
		查询	重填	添加	

ID	业务大类	运营商种类	业务别名	时间	操作
3	联通气象	中国联通	lt	2009-01-14 16:55:48	编辑
2	电信气象	中国电信	dx	2008-11-18 22:13:58	编辑
1	移动气象	中国移动	yd	2008-07-23 07:01:00	编辑

(图 14-9 运营商管理-查询)

单击图 14-9 中的第一个编辑，界面如图 14-10 所示

运营商管理->编辑	
业务大类:	联通气象 *
运营商种类:	2 中国联通 *
业务别名:	lt *
添加时间:	2009-01-14 16:55:48
<div>修改</div> <div>删除</div> <div>返回</div>	

(图 14-10 运营商管理-编辑)

一、电话外呼操作流程

1、管理员通过“电话导入”将号码导入系统，如图 14-11、图 14-12 所示：

电话导入->选择文件	
注意：请选择txt文本文件	
文件名称:	D:\howwe\bin\电话导入 格式.TXT <div>Browse...</div>
<div>导入</div> <div>重填</div>	

(图 14-11 电话导入-选择文件)

电话导入->号码检查			
号码情况:	号码数: 398	重复数: 0	无效数: 0
选择业务			
地区:	日照 (1)		
业务种类:	1 移动气象 (yd)0		
用户状态:	<input type="checkbox"/> 剔除/将回访 (选中时表示次批用户将剔除)		
<div>导入</div> <div>返回</div>			

(图 14-12 电话导入-号码检查)

注意：图 14-12 需选择“地区”和“运营商”，导入后，即可对号码进行回访。

如勾选“用户状态”的“剔除”，导入后将使“待回访”的手机号变为“剔除”。

2、回访员根据“回访单”显示的号码打电话，打完后归类（类的配置在“业务配置”->“业务管理”中进行，其中 11-20 规定为有效的业务、21-26 为无效的业务，可以随意调整；27 为预订，可调整或去掉；28-29 为固定的无效业务，不可调整）。

回访单	
人员编号:	1
手机号:	13512312756
日照	移动
早晚天气	拒绝
早晨天气	未接通
	预订 2009-05
	完成

(图 14-13 回访单)

图 14-13 为回访单界面，其中的“早晚天气”、“早晨天气”可以在“业务管理”处编辑，编辑后请重新登陆。如要定购异地的业务，请选择地区列表中的地区。如要预订，请选择预订的月份后，点击 预订 按钮。

注意：为了维护版权，此页面在 demo 范例中仅为一张图片。

3、管理员对回访员的工作及号码进行管理。

话务统计->查询			
地区:	--请选择地区--	业务种类:	--请选择业务种类--
回访员编号:		用户状态:	--全部用户--
起始时间:	2009-01-01	终止时间:	2009-03-24
<input type="button" value="查询"/> <input type="button" value="重填"/>		<input type="button" value="总数"/> <input type="button" value="导出"/>	
导出文件，右键另存为...			
芝罘区 .0. 待回访 .13512313165 芝罘区 .0. 待回访 .15012345678 芝罘区 .0. 待回访 .13512313162			

(图 14-14 话务统计)

4、回访单产生的有效用户，可以通过“电话回访”->“电话查询”，进行修改。其中预订也属于有效用户。注意，此页面在 demo 范例中仅为一张图片。

回访单修改	
人员编号:	1
手机状态:	处理中
手机号:	13512312756
日照 <small>日照 (1)</small>	移动
早晚天气	拒绝
早晨天气	未接通
预订	

(图 14-15 回访单修改)

5、管理员可以对任意地区的所有类型（用户状态与业务种类的组合）的电话号码进行导出，如图 14-16 所示。

电话导出->查询			
地区:	<small>日照 (1)</small>		
业务种类:	1 移动气象 (yd)0	用户状态:	--全部用户--
起始时间:	2009-01-01	终止时间:	2009-05-03
<input type="button" value="查询"/> <input type="button" value="重填"/>		<input type="button" value="导出"/>	

导出文件，右键另存为...

```
13111131333
13512312312
```

(图 14-16 电话导出)

可将号码以 Excel 格式保存，方便管理。

6、管理员可以对回访员及任意地区的所有类型（用户状态与业务种类的组合）的数进行统计及将电话号码导出，如图 14-17 及图 14-18 所示。

话务统计->查询				
地区:	--请选择地区--	业务种类:	--请选择业务种类--	
回访员编号:		用户状态:	--全部用户--	
起始时间:	2009-01-01	终止时间:	2009-05-03	
<input type="button" value="查询"/> <input type="button" value="重填"/>		<input type="button" value="总数"/> <input type="button" value="导出"/>		

导出文件，右键另存为...

日照 . 0 . 待回访 . 397
日照 . 1 . 处理中 . 1

(图 14-17 话务统计)

话务统计->查询				
地区:	--请选择地区--	业务种类:	--请选择业务种类--	
回访员编号:		用户状态:	--全部用户--	
起始时间:	2009-01-01	终止时间:	2009-05-03	
<input type="button" value="查询"/> <input type="button" value="重填"/>		<input type="button" value="总数"/> <input type="button" value="导出"/>		

导出文件，右键另存为...

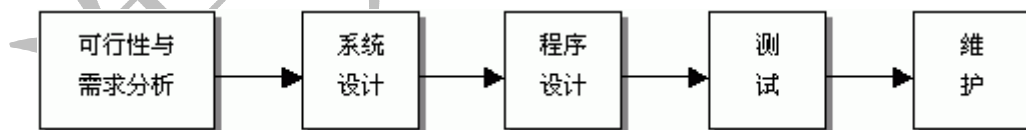
日照 . 0 . 待回访 . 13512313165
日照 . 0 . 待回访 . 15012345678

(图 14-18 话务统计—导出)

第 15 章 HwCALL 系统设计

通过前一章的介绍，我们对电话外呼系统有所了解，如果重新从用户的原始需求开始，我们怎么做，建议先看看“第 21 章 软件工程”，重点看看“21.3 软件方法”，我们先采用轻量级方法来实现电话外呼系统。

轻量级方法步骤如图 15-1 所示，详情请参考“21.3 软件方法”，或使用 Google 搜索。



(图 15-1 轻量级方法步骤)

传统外呼采用电话 + 白纸，即电话号码打印在纸上，员工对纸上的号码进行操作，然后管理员再进行手工统计。

对气象行业，外呼主要用于推广天气预报，即通过外呼，如果用户需要某种天气预报，将这些号码汇总后提交给运营商。运营商确认后，气象局下发天气预报时，用户就能收到。

这就是外呼的原始模型，我们在本章中通过轻量级方法来重新建立一个模型，一个能被计算机识别的模型。

15.1 可行性分析

可行性分析的重点：要解决的问题是什么及此问题是否有行得通的解决办法？

电话外呼要解决的问题：

- 1、协助外呼员外呼，为外呼提供方便；
- 2、减轻管理员的工作量，为管理员迅速提供各种信息。

简单分析一下，这是一个信息管理系统，自然会有很多解决办法。其实随着 IT 的发展，目前几乎不存在没法用计算机来解决的问题，只是代价的大小而已。

15.2 需求分析

这个阶段的任务仍然不是具体地解决问题，而是准确地确定“为了解决这个问题，目标系统必须做什么”，主要是确定目标系统必须具备哪些功能。

电话外呼系统 HwCall 必须具备的功能：

- 1、管理员将号码导入系统；
- 2、回访员根据号码拨打电话，打完后归类；
- 3、管理员对回访员的工作及号码进行管理；

15.3 总体设计-系统设计

如何解决这个问题？首先，应该考虑几种可能的解决方案；然后，推荐一个较好的系统（最佳方案）。

如何解决，软件方面考虑使用什么数据库，用什么语言来编程；硬件方面，该配什么样的电脑。

仔细考虑后，制定几个方案，再和用户沟通，推荐并请用户确定一个方案。

对 HwCall 来说，系统使用 hsql 作为数据库，采用 Java 开发，基于 B/S 结构。

15.4 详细设计-功能模块

应该怎样具体地实现这个系统呢？将系统分成各种功能模块。

HwCall 分为两部分：常用功能和数据管理。分别对应“14.5 用户种类及功能模块”中的一、**功能列表**和二、**数据管理**，详情请查看相关内容，在此不再列出。

第 16 章 HwCALL 编码

根据前面两章的介绍，在了解 HwCall 的基本功能后，确认了功能模块，在这一章我们使用 Java 来实现。

在 15 章的设计中，先搞清楚现实中的业务，建立业务模型，再根据业务模型定义功能模块，这个过程是模型驱动的开发方式，Model-Driven Development，简称 MDD，简单来说就是以建立模型为主要手段的开发。它可应用在软件、系统和数据开发的全过程。

MDD 使用开发人员把更多的精力放在产品的模型和结构上，而不是程序语言和算法实现。UML 是 MDD 概念的一个很好的实现，不过由于 HwCall 的业务简单，所以没用 UML。

在本章中，将介绍 MDD 的基本知识，HwCall 数据库设计及设计代码所涉及的各个知识点。

注：本章仅列一节来说明 Jsp 的使用，便于你留下一个整体印象，如想了解 HwCall 整体代码，请自己查看代码或关注<http://howwe.net/>。

16.1 系统登录详解

一、登录界面

登录界面是使用系统的入口，也是一个最简单的页面，下面详细介绍整个过程。



(图 16-4 登录界面)

图 16-4 所示的页面对应文件为 index.html，完整代码请自己去看，只介绍几个关键点。

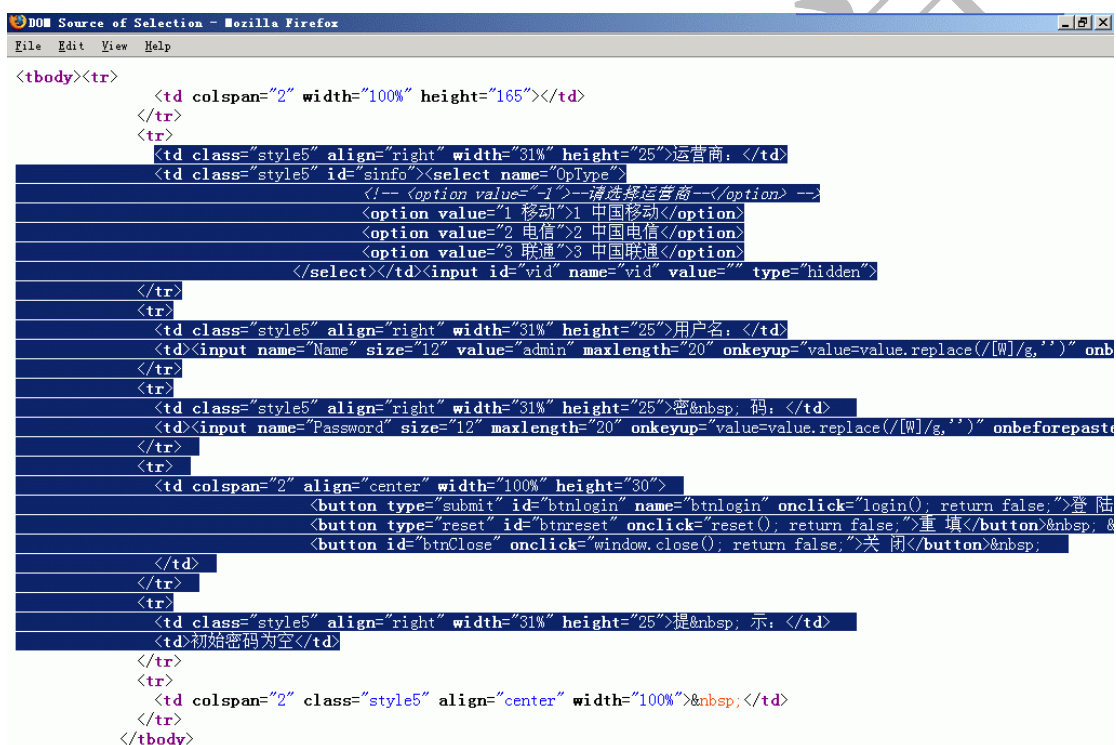
1、在如图 16-4 所示的页面中，真正起作用的只有如图 16-5 所示部分，如果使用浏览器 Firefox，可以如图 16-6 所示，先选中想要查看源码的区域，单击右键，再选择“View Selection Source”来查看源码，显示的源码如图 16-7 所示，内容不做解释，不懂的地方请自己去 Google。唯一要注意的是，这种方式下显示的代码是 Firefox 分析后的代码，可能与现有的代码不一样，可对比一下如图 16-8 所示的“Eclipse 中的相应代码”。



(图 16-5 登录界面)



(图 16-6 查看登录界面核心代码)



(图 16-7 Firefox 显示的登录界面核心代码)

(图 16-8 Eclipse 中的相应代码)

```
<form id=fdata method='post' action='indexLogin.jsp' name="LoginForm"
onSubmit="return CheckForm(this)">
```

二、登录处理 indexLogin.jsp

```
String name = request.getParameter("Name");
String password = request.getParameter("Password");
String opType = request.getParameter("OpType");
```

```
SmsDaoLogin loginDao = new SmsDaoLogin();
HashMap loginMap = loginDao.login(name, password);
```

```
if (loginMap == null) {
    %>用户名或密码出错，请重新登陆。<a href="index.html">返回登陆</a>
    <meta http-equiv="refresh" content="3;URL=index.html">
    <%
```

```
} else {  
    //详细登录处理，暂时略  
}
```

从以上代码可见，如果登录失败，将自动重新登录，否则获取登录信息。

4、获取登录信息

```
String moduleIdList = "", otype="";  
moduleIdList=loginMap.get("ModuleIdList").toString();  
admin.add("ID", loginMap.get("ID").toString());  
admin.add("name", name);  
admin.add("OpType", OpType);  
admin.add("ModuleIdList", moduleIdList);  
//使用session控制登录，不过admin也是session  
session.setAttribute("qx", moduleIdList);  
  
otype=loginMap.get("tInfo").toString();  
otype="" + otype;  
admin.add("tInfo", otype); //业务种类  
otype=loginMap.get("dInfo").toString();  
otype="" + otype;  
admin.add("dInfo", otype); //地区  
otype=loginMap.get("uInfo").toString();  
admin.add("uInfo", otype); //用户状态  
  
otype="1, 移动*, 2, 电信*, 3, 联通*";  
admin.add("uoInfo", otype); //运营商字符串  
otype=loginMap.get("udInfo").toString();  
otype="" + otype;  
admin.add("udInfo", otype); //地区字符串  
admin.add("Info", loginMap.get("Info").toString()); //用户状态字符串  
  
otype=loginMap.get("otype").toString(); //用户类型：管理员、回访员
```

重要代码后面都有说明，就再详细说明。

5、根据用户类型，导向不同的页面

```
if (!otype.equals("0")) { %><meta http-equiv="refresh"  
content="1;URL=content.htm">  
<%> else { %><meta http-equiv="refresh" content="1;URL=smsCall.jsp">  
<%> %>  
<p>登陆成功，正在进入系统...</P><%>
```

注意：在这里使用的是 meta，也可以使用其他方式：如 response.sendRedirect、js

脚本。

三、login 数据操作

login 是数据操作类 SmsDaoLogin 的一个方法，SmsDaoLogin 在数据操作包 net.howwe.core.dao 下，对应文件为 SmsDaoLogin.java。数据操作的基本原理在“16.5 数据层介绍”有简单介绍，再详细说明一下。以下是 login 方法的全部源码：

```
public HashMap login(String name, String password) throws Exception {
    HashMap mapResult = null;
    ResultSet rs = null;

    Dw dw=new Dw();//使用Dw
    try { //使用Dw
        dw.sqlPs(Login_SQL);
        dw.setString(1, name);
        dw.setString(2, password);
        dw.sqlRs();
        rs=dw.rs; //->使用Dw

        if (rs.next()) {
            mapResult = new HashMap();
            mapResult.put("ID", rs.getString("ID"));
            mapResult.put("ModuleIdList", rs.getString("ModuleIdList"));
            mapResult.put("otype", rs.getString("result"));

            SmsDaocOpDecl recDao=new SmsDaocOpDecl();
            mapResult.put("tlInfo", recDao.getInfo());
            SmsDaocOrderType recuDao=new SmsDaocOrderType();
            mapResult.put("uInfo", recuDao.getInfo());
            mapResult.put("Info", recuDao.getuInfo());
            SmsDaocOrderDept recdDao=new SmsDaocOrderDept();
            mapResult.put("dInfo", recdDao.getInfo());
            mapResult.put("udInfo", recdDao.getudInfo());
        }
    } catch (Exception ex) {
        log.error("", ex);
    } finally {
        dw.close();//使用Dw
    }

    return mapResult;
}
```

1、入参及返回对象

先看方法的定义：HashMap login(String name, String password)，入参有两个 name 和 password，返回对象为 HashMap。使用 HashMap 的原因是返回的参数超过一个，否则就可

以直接返回。至于 HashMap 是什么，自己去查 JDK 的 API，其实多看看 API 效果很不错。

2、登录判断数据操作

使用 JDBC 包装类 Dw，其实操作只是一个查询，SQL 语句为 *Login_SQL*: `select ID, ModuleIdList, result from tSmsbOperator where name=? and password=?`，但操作时没对入参进行 SQL 注入检查。

浩为语：先学会走路，再去写如何跑步，如果一开始就去跑步，只会摔跟头。

具体代码如下：

```
Dw dw=new Dw();//使用Dw
dw.sqlPs(Login_SQL);
dw.setString(1, name);
dw.setString(2, password);
dw.sqlRs();
rs=dw.rs; //->使用Dw
```

3、登录成功后获取相关信息

第2步提交查询后，如果存在相关用户，则先获取用户信息，再获取系统所需的其他信息。

```
if (rs.next()) {
    mapResult = new HashMap();
    mapResult.put("ID", rs.getString("ID"));
    mapResult.put("ModuleIdList", rs.getString("ModuleIdList"));
    mapResult.put("otype", rs.getString("result"));

    SmsDaocOpDecl recDao=new SmsDaocOpDecl();
    mapResult.put("tInfo", recDao.getInfo());
    SmsDaocOrderType recuDao=new SmsDaocOrderType();
    mapResult.put("uInfo", recuDao.getInfo());
    mapResult.put("Info", recuDao.getuInfo());
    SmsDaocOrderDept recdDao=new SmsDaocOrderDept();
    mapResult.put("dInfo", recdDao.getInfo());
    mapResult.put("udInfo", recdDao.getudInfo());
}
```

4、返回 HashMap

indexLogin.jsp 对调用 login 返回的 HashMap 进行处理，根据结果引导下一个页面。

第 17 章 IT 更需要引导

有人说软件开发最困难的不是技术，而是交流；软件开发的价值在于开发者通过做出来的软件，来解决用户的问题；只有解决了用户的问题，他的价值才能体现，否则他就会有挫败感。可是当你对这些一满足，就会失去方向，这是很多牛人一到职业生涯的某些阶段，尽

管技术很牛，却还是迷茫的原因。

你现在想做什么？必须做什么！你以前做过什么？现在能做什么！以后要做什么？现在怎么做！个人价值在人生引导中只是很小的一块，因为个人价值只体现在个人发展中。

1、个人的发展需要教育，更需要引导；企业的发展需要人力资源，更需要自我发展。教育怎样才能把两者结合起来，学以致用是根本。

2、个人发展存在很多不可控因素，道法自然让人坦然面对。

3、资源整合，对个人和企业同等重要，学习型组织更体现了对个人的资源整合。

4、从系统论的角度来看：人生不会有所谓的最终的成功，有的只是你对人生的豁达和你看事物的那份坦然。人活的是一份良知和一份责任，活的是一种平凡和一种心态！

17.1 IT 培训重在引导

在QQ群里有人贴了一篇CSDN上的文章，标题“培训机构出来的学员不适用，我们需要怎么样”，http://cto.csdn.net/Hot_Discuss.aspx?Name=sunshanming&pointid=1410，提到（原话）：“我发现了一个有意思的事情，所有的培训机构都号称课程面向实战项目，但仔细看看那些项目就会发现，不过是些玩具项目，什么网上商城、图书管理等等，名头很大，内容少得可怜，往往是几个人一两个月就做出来的，而且都是按照规定的模式，实现几个技术点就完了。询问起来，一个人都是负责其中一部分，对整体项目都不了解，怎么看都像是一个玻璃杯里用营养药水培育出来的花朵，一到真实生存环境里，必死无疑。”

曾看过几本培训机构的教材，初一看，内容真多，可仔细一想，半年真能教那么多吗？2009年2月，浩为新进了一名在某上海知名培训机构全日制培训过一年半的员工，本期望他能迅速融入Java开发团队，但上班第一天就不得不对他重新培训，因为他除了能写一些简单的Java代码之外，连Web项目的运行原理都不懂，原来没人教过。通过两个星期的培训，他感慨万分：“如果早有这个机会，就不用浪费一年半时间及3万多培训费了”。

2009年3月底，我曾和某国内排前5的培训学校的人力总监交流过，有句话最有印象：“现在培训机构除了牌子不同，其他没啥不同”。

参加过9月25日的蓝海创投“早中期项目专场对接会”第二季，不为投资，主要是请投资人提建议。当时只准备了四页纸，一般人只给两页纸，第一页为“浩为成长计划”的宣传介绍，第二页是《喻世明言新篇-人生需要引导》中的“1.2 我要喝水”，第三页“浩为简介”，第四页“老虎基金4000万美元注资培训机构学而思”。给我印象最深的是蓝海创投的副总，他听我介绍完，再一看资料上的介绍：“浩为成长计划：多层次Java技能培训、个人定位分析、培养思维习惯”，就对我说：“你这是挂狗头卖羊肉，别人简单一看，以为你是搞Java培训的，其实你的重点是人生启迪，也就是你所说的人生需要引导……”看来我以前对以人生引导为主、还是以Java培训为主的犹豫很正常，从长期看，当然得以人生引导为主，但从短期来看，先得通过Java培训积累现金。

人生需要引导，以多层次Java培训为起点，知道自己要做什么：个人定位分析、思维习惯培养、职场体验等，更提供参与有偿项目、领导力锻炼营的机会，详情howwe.net。陈聚雄从“李开复给大学生的公开信”中找到了前进的方向，加入浩为QQ群后，通过努力再成为浩为员工，人生需要引导将为参与者提供更多机会。

《从人生需要引导论Java快速开发》作为Java培训的专用教材，培训内容：Java入门培训（免费培训该书前13章）、Java基础培训（培训全书27章），Java项目培训（多项目完整实现），更提供在培训就参与外包项目的机会。教是为了不教，学以致用，引导学更

激发你的潜能。该书主要针对准备从事 IT 的初学者及 Java 入门者，既能让初学者快速入门、掌握实用技术，又能让入门者迅速树立整体思想。更让所有读者从根本上明白自己需要什么，懂得资源整合、学以致用，快速开启财富之门。

编程的核心是思想，语言只是思想的一种实现方式而已，编程基础：数据类型、控制结构、数据结构。当你有了思想，即使不懂某种语言，但你很快就能用其实现功能。在这思想的指导下，从没接触过 PHP 的陈聚雄十天就能参与项目开发，详情见 howwe.net。

以下是培训相关的一些观点：

1、现有培训机构的培训课程能否真正面向实战环境，找真正做过项目的教师来研究出一套合理的课程，面向具有语言基础的学员，作为项目实践提高班，项目的选择要贴近实际，不要从书本上随便找两个例子，那不是现实，而且项目不能太小，10 万行以下的项目就没多大意义，因为足够大的项目才能发现很多问题，遇到面试官的时候，一两句话就能看出你的项目经验。真心希望培训机构早些开发出这样的课程，使我们的学员真正具有实战经验。

2、因为培训和教育基础的脱节，培训不看实际的就业使用情况，而是看学生学习的兴趣程度，学生觉得简单易学，可能就愿意学；学生觉得难时，培训机构就很麻烦；而面向本科或者大专学历的培训又不被认可，造成了培训内容和工作需要的脱节现状。

3、现在商业化太严重，没有扎实的基础知识和经验，仅仅几个月培训，掌握现在纷繁复杂的框架是不可能的。实践才是最好的课堂，很多时候我劝那些想去参加培训的朋友，不如不要钱去一个公司参加项目。

4、企业对刚从学校出来的学生进行培训是必须的。企业需要的是专才，需要在某些方面有突出能力的人，而学校培养更多是培养通才，让这些学生有更多的机会入职。不同的要求造成了矛盾。

5、软件开发最困难的不是技术，而是交流。如果培训在这方面能够更接近实际的话，那么培训机构就会更有竞争力。一个人要发展，首先要认识到自己的价值。如果一个人连自己的价值都不清楚，他朝什么方向发展？做软件开发的人，他的价值在于通过做出来的软件，能够解决用户的问题。解决了用户的问题，他的价值才能体现。否则他就会有挫败感。

6、你现在想做什么？必须做什么！你以前做过什么？现在能做什么！以后要做什么？现在怎么做！这个是内在的，也是最重要的价值所在。不然，就算你因满足客户后有成就感；但当你这些都习惯了，你就会失去方向，这也是很多牛人到了职业生涯某个阶段时都觉得技术很牛却还是迷茫的原因。培训机构缺乏的正是价值实现的途径。

7、培训中心也许可以做到全真项目开发，做到公司制度考勤！但是即使这些真做到了，培训中心又怎么能让更多的学生感觉到自身的责任。有人说学校只是学校，最终是以宣传和盈利为目的。其实如果学校真以宣传和盈利为目的，就要严格要求学员，让学员学到真本事。否则培训完的人大部分都有一种被欺骗的感觉。

8、培训中心不管如何宣传，如何夸大！只需要明白一点就行，“师傅领进门，修行在个人”。所谓“师傅”就是培训机构，不可否认，个人修行还需明灯引导！

浩为培训的根本原则：个人价值在人生需要引导中只是很小的一块，因为个人价值只体现在个人发展中。

1、个人的发展需要教育，更需要引导；企业的发展需要人力资源，更需要自我发展。教育怎样才能把两者结合起来，学以致用是根本。

2、个人发展存在很多不可控因素，道法自然让人坦然面对。

3、资源整合，对个人和企业同等重要，学习型组织更体现了对个人的资源整合。

4、从系统论的角度来看：人生不会有所谓的最终的成功，有的只是你对人生的豁达和

你看事物的那份坦然。人活的是一份良知和一份责任，活的是一种平凡和一种心态！

17.2 引导学思想开创 IT 培训新模式

2009 年 2 月，浩为新进了一名在某上海知名培训机构全日制培训过一年半的员工，本期望他能迅速融入 Java 开发团队，但上班第一天就不得不对他重新培训，因为他除了能写一些简单的 Java 代码之外，连 Web 项目的运行原理都不懂，原来没人教过。通过两个星期的培训，他感慨万分：“如果早有这个机会，就不用浪费一年半时间及 3 万多培训费了”。

可想而知，社会还有多少受过或正在接受这种培训的人，也就是不少培训机构培训出来的软件蓝领，难道他们乐意于只会写代码？这位兄弟当初是抱着学好 Java 的愿望去的，用心学过之后，却没学到实用的本领。

2009 年大学毕业生人数为 610 万，很多人“被就业”，可就业情况还是不乐观。另一方面，企业却招人难，即使招了人也没法用，必须重新培训，但接受某些培训机构的培训之后，情况还是如此。

有人总结出大学四年生活路线：大一迷茫、大二寻觅、大三分化、大四就业，很多同学找工作受挫后，选择回家等待安排，更多时间就在宿舍玩电脑、游戏、看电影。很多人根本不去想如何提高自己，让自己有更大的优势去找工作。还有一些人想努力却无从下手。

教育的目的是什么，很多人在思考，更有不少人在反思。下图为人生需要引导（简称引导学）相关概念的思维导图：



(图 17-1 “人生需要引导相关概念”思维导图)

- 1、个人的发展需要教育，更需要引导；企业的发展需要人力资源，更需要自我发展。教育怎样才能把两者结合起来，学以致用是根本。
- 2、个人发展存在很多不可控因素，道法自然让人坦然面对。
- 3、资源整合，对个人和企业同等重要，学习型组织更体现了对个人的资源整合。
- 4、从系统论的角度来看：人生不会有所谓的最终的成功，有的只是你对人生的豁达和你看事物的那份坦然。人活的是一份良知和一份责任，活的是一种平凡和一种心态！

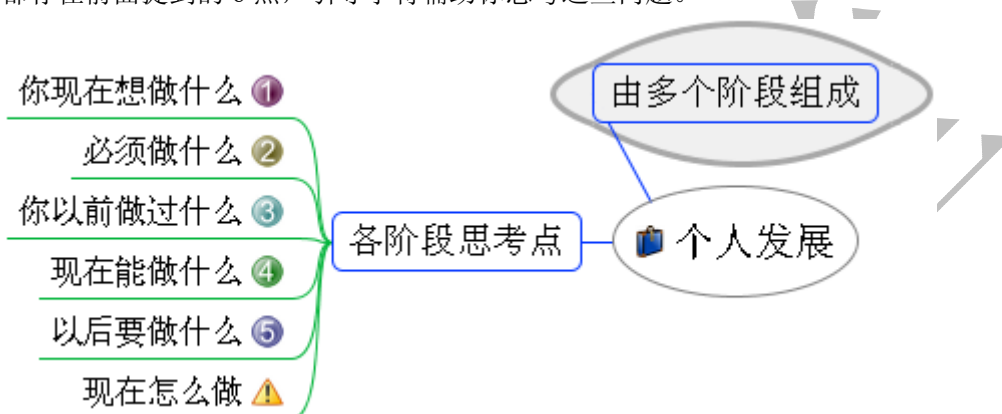
人生需要引导，以 Java 快速入门及项目开发培训为主、结合项目外包，给参与者更多实战机会；以引导学为辅，引导参与者在培训后进入社会更加知道自己要做什么，通过日程事务管理，携手共进，共同发展；为社会培养一批写作能力强、能说会道、懂技术的人，更为浩为储备一批人。

编程的核心是思想，语言只是思想的一种实现方式而已，编程基础：数据类型、控制结

构、数据结构。当你有了思想，即使不懂某种语言，但你很快就能用其实现功能。在这思想的指导下，从没接触过 PHP 的陈聚雄十天就能参与项目开发，详情见 howwe.net。

人生需要引导，以多层次 Java 培训为起点，知道自己要做什么。《从人生需要引导论 Java 快速开发》作为 Java 培训的专用教材，培训内容：Java 入门培训（免费培训该书前 13 章）、Java 基础培训（培训全书 27 章），Java 项目培训（多项目完整实现），更提供在培训就参与外包项目的机会。教是为了不教，学以致用，引导学更激发你的潜能。

你现在想做什么？必须做什么！你以前做过什么？现在能做什么！以后要做什么？现在怎么做！认真考虑考虑这些问题，只会对你百利而无一害。个人成长可以分为多个阶段，每个阶段都存在前面提到的 6 点，引导学将辅助你思考这些问题。



(图 17-2 “个人发展思考点”思维导图)

人有偷懒的天性，面对漫长的人生之路，很多人的失败就在于不能持之以恒。有人说，优秀是种习惯。引导学结合思维导图 FreeMind 等的使用，并辅以日程事务管理，培养你的思维习惯，期待你早日走上成功之路。

社会需要什么、企业需要什么、你需要什么，还没有正确定位的你，该怎样确定自己的方向？恰当的定位最需要自我改变，但自我改变是个艰难的过程，需要外力的协助。引导学给你外力，更将一帮人组成群体（浩为资源堂 zyt.howwe.net），携手共进。

浩为资源堂整合浩为学堂，推出浩为百校送书行动，高达万册之多。主要针对浩为资源堂（zyt.howwe.net）的注册用户，通过自己的努力赚取积分，用户可用积分兑换以下礼品：浩为学堂提供的一个月“职场技术组”培训（1000 积分，含 Java 入门培训）、“浩为资源堂月刊”电子档（0 积分）、“浩为资源堂特刊”印刷品（1000 积分）、《从人生需要引导论 Java 快速开发》入门版《数据库开发这点事-Java 快速入门》或《喻世明言新篇-人生需要引导》印刷品（均 2000 积分）、《从人生需要引导论 Java 快速开发》印刷品（5000 积分）等，同时为用户提供高校联盟板块组织的实地活动：IT 学习交流、公益活动 等。印刷品需对方支付邮资，因为免费的资源很多人从不珍惜。

浩为认可的能力包括：做事态度、思维能力、写作能力、口头表达能力，并以此培养 QQ 群管理层。陈聚雄是管理层中的佼佼者，“集美大学”学前教育专业，大学前两年很迷茫，从“李开复给大学生的公开信”中找到了自信和方向，自学计算机编程，2009 年 5 月加入浩为 QQ 群，多次整理或领导整理 QQ 群聊天记录，有多篇文章入选“人生需要引导”系列，毕业后进入浩为，目前正参与或组织浩为-项目群多个项目的开发，其中有上海华东师范大学的两本软件工程教程。除陈聚雄之外，管理层中还有多人将进浩为实习，甚至安排工作。

人生需要引导，以多层次 Java 培训为起点，知道自己要做什么：个人定位分析、思维习惯培养、职场体验等，更提供参与有偿项目、领导力锻炼营的机会，详情 howwe.net。陈

聚雄从“李开复给大学生的公开信”中找到了前进的方向，加入浩为 QQ 群后，通过努力再成为浩为员工，人生需要引导将为参与者提供更多机会。

相关内容：

- | | |
|-------------------|---|
| 1、从小儿推拿看人生需要引导 | http://howwe.net/plan/82.html |
| 2、从“我要喝水”谈引导的迫切性 | http://howwe.net/plan/47.html |
| 3、从喻世明言谈人生需要引导 | http://howwe.net/plan/84.html |
| 4、从网瘾谈人生需要引导 | http://howwe.net/plan/81.html |
| 5、从李开复离职创业谈人生需要引导 | http://howwe.net/plan/8.html |

17.3 引导学概述：人生需要引导

由于教育的局限，很多人迷失了自我；再加上父母等长辈对孩子的溺爱，太多的人迷失了本性，以为自己得到的一切都是理所当然的，以致忘了感恩、珍惜及努力。他们一进入社会，很多人碰到头破血流，有些人心甘堕落甚至犯罪。因为社会更多的是利益的交换，是一个市场，你拿不出别人想要的东西就没法得到自己想要的东西。太多的人由于没能掌握真正的谋生手段，只能通过堕落或犯罪来维持生活。

人生需要引导，简称为“引导学”，在于找回人生的本质，让人知道自己要做什么。只有自己想要的东西才会珍惜，受过霜的柿子才会甜，人也只有经过磨炼才会成熟。人生需要引导系列的撰写，希望更多的人能从众多经历教训中少走弯路，减少成长中的磨难，让自己的生活变得实在及精彩。

随着中西交流的发展，西方思想被更多的人所了解。在教育方面，西方讲究自立，中国重在孝顺，但在这方面也做得极差。有人说教育需要中西结合，西方的自立和中国的孝顺都很重要，但两者有矛盾、甚至冲突。其实两者没有任何矛盾，孝顺更需要自立，不能自立，你拿什么来孝顺？你真正自立了，父母才不会天天为你操心，这才是对父母的最大孝顺。

有人说父母对子女的包办、溺爱害了小孩，父母的很多做法只是适得其反。因为父母不知道自己该怎么做，所以他们用自以为最好的方式去教育小孩。随着网瘾标准的炒作及野蛮治疗事件的曝光，越来越多的人“被网瘾”。有些家长被吓怕了，将自己的子女送货上门，成了网戒中心的赚钱对象。却不知，父母耗费了大量的冤枉钱不算，子女却被扭曲了心灵，带来了精神和肉体的双重伤害。

有人说很多人都是无证上岗，即没有人指导就恋爱、结婚、生子等。除了极少数人先知先觉、无师自通之外，其他人大多要走弯路，甚至造成遗憾或悲剧。人与人的相处，尤其是父母与子女的相处、恋爱婚姻中两性的相处存在同样的问题。由于房价的高企，住房成为家庭的一大问题，使得很多子女成年后还得和父母住在一起，但子女成长时的教育就决定了以后能否和谐相处，可初为人父母时该如何教育小孩呢？

人生需要引导系列的撰写，就是对这些问题的思考，也是对如何教育 5 岁侄女的思考，更是从业 10 年、创业 7 年的总结。有人说作者喻桃阳人生经历十分特殊，历经磨难，属兔的双鱼座，加上经历过一些事情，更多一丝敏锐及洒脱，看透了不少事情，比常人更多一份坦然及专注。作者 2000 年上海东华大学毕业，专业信息管理系。

作者从系统论的角度，以信息管理的方式来思考人生中的很多问题。简单来说，人就是一个系统，人生中的事情可以简化成不同的信息，用信息管理的思想去处理人生的一切。引

导学按照人的成长过程，分成多个阶段，每个阶段都存在以下问题：你现在想做什么？必须做什么！你以前做过什么？现在能做什么！以后要做什么？现在怎么做！

人生需要引导，道法自然是根本，感恩成就人生、态度决定一切、激发你的思维潜能、知道自己要做什么、争当领头人、资源整合、IT 简单化、相处的艺术从多个角度阐述人生中的注意点。引导学更包括养生保健方面的引导。有人说《黄帝内经》利用阴阳五行以及五脏六腑理论，引导人们如何顺应四季变化来生活；描述了许多疾病的起因，根据人体经络阐明了各种疾病的治疗方法。一件别人眼中微不足道的小事，却极大影响了你，特别是心情，而长时间的心态失常则会引起身体的不适。对于身体的不适，西医讲究头痛医头脚疼医脚，却不去寻找不适的原因。所以人需要了解人体保健知识，才能使身心同时保持健康。

人生三重境界：看山是山，看山不是山，看山还是山。这就是所谓：人本是人，不必刻意去做人；世本是世，无须精心去处世，这才是真正的做人与处世。

绝大多数人追求一生、劳碌一生、心高气傲一生，最后发现自己并没有达到自己的理想，世界依然是那个世界，于是抱恨终生。可有些人通过反思及提炼，茅塞顿开、回归自然，开始专心致志做自己该做的事情，不与旁人有任何计较，任你红尘滚滚，我自清风朗月。这个时候，人就进入第三重境界，看山又是山，看水又是水了。

从系统论的角度来看：人生不会有所谓的最终的成功，有的只是你对人生的豁达和你看事物的那份坦然。人活的是一份良知和一份责任，活的是一种平凡和一种心态！中国有句古话“盖棺论定”，这也是对中国几千年来成功人士的一个概括和告诫，如果你真想追求那种最终的成功，就脱离不了“盖棺论定”的宿命。

一教授推荐我多去看看某管理大师的著作，我一看简介：一堆学历及头衔，我再看某书的目录，就懒得再看。我说：“只能说×是一个写手。”对方回复“或许是吧，他的有些书我也看不下去。他的思路跟你比较，貌似有点走偏，把传统文化里的‘权谋机巧、人心叵测’做了大发挥，比较适合年纪大的人看，你的书主要在告诉读者走正道”……“权谋机巧、人心叵测”通俗一点讲就是尔虞我诈，工作 30 多年加上凡人望尘莫及的经历，绝大多数人只能望洋兴叹。而引导学，讲究道法自然，大多是大众化的内容，就连那位教授都向我报喜：“刚才我走开一会，8 岁的女儿在我电脑上看你写的东西，看得津津有味。”《喻世明言新篇—人生需要引导》精选系列小故事，挖掘隐藏其中的哲理，使小孩也喜闻乐见。

国庆阅兵时，我在看涂学能的《开国大阅兵》，唯一有印象的是林彪打仗的独断。有人说林彪是一个统兵打仗的怪才，两耳不闻窗外事，一心只为打胜仗。林彪打仗经常直接指挥到师，运动战、重要战斗或重要方向，还直接指挥到团。部队行军到达指定地域，第一件事就是报告当时敌情。三下江南时，林彪要求 20 分钟内汇报敌情。一仗下来，林彪 6 小时内要简报，24 小时要详报，逼着部队及时总结经验教训，就这样造就了林彪的战神地位。

引导学提出为人处事的注意点，阐述资源整合的必要性，倡导“IT 简单化”；结合思维导图的使用及范例，提出思维可以条理化，问题的解决也有章可循。通过分析 Google 企业战略，描述当今互联网格局，阐述了公司战略的重要性，更阐明了人生需要引导的重要性。

17.4 人生需要引导系列总序

喻桃阳自序

《从人生需要引导论 Java 快速开发》，简称《人生需要引导》，作为浩为“人生需要引

导”系列丛书首册，提出需要引导的五大对象：大学生、家庭、企业、在职人员、中小學生。人生需要引导，简称“引导学”，致力于解决众人不知道自己做什么、而企业却招人难的问题。希望能将更多的人引导成才，他们再去引导更多的人，因为这个社会需要引导的人太多太多。

为啥是五大对象，从“自我经历：知道自己做什么”可以看到，作者本人人生经历特殊，历经磨难，1975 年出生湖南湘乡，2000 年毕业于上海东华大学，专业信息管理系；从 2003 年开公司以来，创业长达七年，比常人更能体会到人生需要引导的重要性。所以内容从单纯写技术开始发展到技术和人生结合，发展到一系列丛书。

《人生需要引导》由技术及人生两部分组成，技术部分主要针对准备从事 Java 的初学者，既能让初学者快速入门、掌握实用技术，又能让入门者迅速树立整体思想。更让所有读者从根本上明白自己需要什么，懂得资源整合、学以致用，快速开启财富之门。

浩为采用参照现有项目，自己动手做项目；通过通俗易懂的描述，图文并茂，带你畅游数据库。代码只是一个过程、编程是一种思想、项目开发更是一个系统工程，浩为先讲述过程、塑造你的思想，让你树立系统的理念，从而实现你、实现浩为的目标。

有人说该书是本无厘头书，东讲一段，西讲几句，不知所云。如果一味固守自己的观点，我建议你走开，该系列不适合你，因为很大篇幅讲的是思想，解决问题的思路。

系列之二：《喻世明言新篇-人生需要引导》，简称《人生需要引导 II》，作为《人生需要引导》人生部分的概括及延伸，主要针对学生及在职人员，提出为人处事的注意点，并阐述资源整合的必要性。感恩解决生存问题，态度及要做什么解决发展问题，资源整合解决壮大问题，相处的艺术解决相处问题，道法自然解决家庭问题。

系列之三：《数据库开发这点事》，重点讲述系统开发案例，全书结合 Java、C++、PHP 等语言，讲解 jQuery、HoCAS、Mina、phpBB 等框架，结合各种数据库的使用，使 IT 人士具备快速开发高效应用系统的能力。并延伸为多本快速入门，分别为《数据库开发这点事-Java 快速入门》、《数据库开发这点事-PHP 快速入门》、《数据库开发这点事-C&C++快速入门》。

浩为到底想干嘛，其实也只是围绕“人生需要引导”这个范畴，做一点小事而已。在 QQ 群里叫“浩为”而拒绝别人称老师，一是因为听别人称自己为老师就感到别扭，二是因为自己仅仅是一个商人，称“浩为”算是一种商业操作吧。

以后如果能在我墓碑上刻着：“勤于思考的商人在此长思”，我就会特别开心，因为其他仅为虚无。有人说属兔的双鱼座，加上经历过一些事情，更多一丝敏锐及洒脱，看透了不少事情，比常人更多一份坦然及专注。浩为致力于人生需要引导的推广及浩为资源堂的壮大，期待能让更多人的生活变得实在及精彩。

已经在构思的书籍：

1、细节决定成败，主要面向中学生。该书将综合个人、企业、婚姻、家庭、保健等多个方面，以小见大，阐述成功的细节。道生一、一生二、二生三、三生万物，然而道法自然，该书从小故事出发，引出深藏其中的道理。

2、相处的艺术，家庭成员如何相处，团队成员如何相处，上下级如何相处等。

3、和小孩讲道理，很多人在小孩教育上，无从下手。为了 5 岁侄女的健康成长，作者在思考，更在实践。时常在与侄女的交往中，发掘能培养她能力的小事，更对她的不足，大棒加胡萝卜，目前已取得不少成绩。

题外话：如何教育孩子，多去思考

一把 1 块钱的水枪，我却能让我侄女用来在地上写字，和我比赛在墙壁上看谁打出的水迹最长。我曾把水枪拆了，待她哭完后，教她如何装。有次我妈对我讲，这侄女很自豪地说：“奶奶，我会拆装水枪了”。有一次我演示给侄女看，水还能从进水管那头打出来。

17.5 作者：喻桃阳简介

喻桃阳，浩为创始人、“人生需要引导”系列丛书作者、“引导学”发起人，1975 年出生湖南湘乡，2000 年毕业于上海东华大学，专业信息管理系。坚信感恩成就人生，倡导资源整合、学以致用。2008 年 3 月 29 日，父亲突然去世，尚未走完人生旅途的 74 周岁。2009 年 3 月 30 日我开始撰写该系列，以此纪念我的父亲。

在书中，不用“喻桃阳”，而只用“浩为”，因为将浩为公司做大是我的人生追求。在“活着是一种责任，多做善事”中提到“唯一感到遗憾的是，没有在我爸去世前，把公司做大”，所以我用心写书，因为写出一本能给很多人带来帮助的书是我的追求。

有段时间曾和朋友吹牛说自己啥都懂一点、但啥都不精，因为看过很多书，清楚众多名人的故事及不少学科的基础，甚至对中医经络、易经八卦、面相风水、道教佛教思想等都有了解。常和别人开玩笑，我是一个幼儿园都没毕业的文盲。有人却说作者本人人生经历十分特殊，历经磨难，属兔的双鱼座，加上经历过一些事情，更多一丝敏锐及洒脱，看透了不少事情，比常人更多一份坦然及专注。看过我手的人，只要懂一点手相的就会说：“看来你饱经沧桑”，希望我的经历能帮助更多的人。

从 2003 年开公司以来，作者创业长达七年，比常人更能体会到人生需要引导的重要性。所以内容从单纯写技术开始发展到技术和人生结合，发展到一系列丛书，提出需要引导的五大对象：大学生、家庭、企业、在职人员、中小学生。

浩为到底想干嘛，其实也只是围绕“人生需要引导”这个范畴，做一点小事而已。在 QQ 群里叫“浩为”而拒绝别人称老师，一是因为听别人称自己为老师就感到别扭，二是因为自己仅仅是一个商人，称“浩为”算是一种商业操作吧。以后如果能在我墓碑上刻着：“勤于思考的商人在此长思”，我就会特别开心，因为其他仅为虚无。浩为致力于引导学的推广及浩为资源堂的壮大，期待能让更多人的生活变得实在及精彩。

为何要写这一系列丛书？

其一，该系列从作者父亲逝世周年纪念日第二天，即 2009 年 3 月 30 日开始撰写，以此纪念作者的父亲。

其二，该系列是浩为公司发展战略的一个组成部分，实现浩为百校送书的前提。该行动预计给全国 100-200 所大学学生送书高达 10000 册，详情请参考“浩为百校送书行动”。

其三，浩为想做点事，也有这个能力去做事，所以一直很努力，该系列是浩为最需努力的一个方向。多层次 Java 培训的推出，是“人生需要引导”系列丛书的产业化。

早在 2001 年，有天我头发理得很短，一进公司，小组长就说普京来了，我很奇怪。他走过来一解释，说我像普京（俄国的普京），这是第一个当面说我长得像普京的人。



以上是对照图，不少朋友都说我像普京，但我觉得不像，鼻子和额头就不像，眼睛更不像。既然朋友们说我像，就借普金为自己宣传宣传吧。人生需要引导，知道自己要做什么，多层次 Java 培训，人生需要引导，期待能让更多人的生活变得实在及精彩。

17.6 学以致用开创大学社区新模式

摘自浩为资源堂<http://zyt.howwe.net/>

2009 年 11 月初，我出差北方，在某省会城市转车，从火车北站经火车站换车，早上 6 点多一下火车，好冷，后得知是郊区在前一天下雪了。更好玩的是出租车拒载，需多人拼坐才载客，并且按人头收。

在浩为 QQ 群 78928780 与人交流时，有人说“这是份子钱给逼的，出租车一拼坐，同样的油耗挣更多的钱”，我说：“其实城市最忌讳这个，尤其是投资人”，为什么呢？投资人讲究的是收益，如果一个城市的交通都让人反感，对大多数人来说，即使他可能会赚很多钱，但他的员工在这个城市没有安全感，他的外来客户一到这个城市就没安全感，他何以为继？

浩为资源堂 <http://zyt.howwe.net> 能给人带来什么呢？我们先看其定位：浩为资源堂，你的生活因此而精彩；学以致用，以 IT 技术、职场体验为主，面向大学生的个性化社区式资源圈。

不论是技术，还是职场经历，最终都要归于学以致用。从人的成长过程来说，读书、工作、生活，也要落到“用”字。读书为了什么？工作为了什么？生活为了什么？

有人说“读大学是为了挣大钱、娶美女”，更可悲的是有人以为这是一种“思想解放”。不少人带着这种低级的功利心，大学被曲解为赚钱的职业培训机构，人被制造一堆合格的赚钱机器！然而，人终究不是机器，人是有感情的，也是有兽性的。有些人资产上千万，可是为了得到更多，不择手段，甚至害人害己。

人到底为了什么而活着？一个人的成长有三个层次：生理上的，他的身体发育成熟了；法学意义上的，他有了选举权和被选举权；精神上的，他有属于自己的价值尺度并自觉地用它来约束自己的行为。只有精神上的成熟，他才会懂得自律、负责任、珍惜、敬畏，懂得不能让自己所爱和自己爱的人操心，懂得自己的命运掌握在自己的手里。

人生需要引导，简称为“引导学”，在于找回人生的本质，让人知道自己要做什么。只有自己想要的东西才会珍惜，受过霜的柿子才会甜，人也只有经过磨炼才会成熟。人生需要引导系列的撰写，希望更多的人能从众多经历教训中少走弯路，减少成长中的磨难，让自己的生活变得实在及精彩。

浩为资源堂是一个面向大学生的个性化社区式资源圈，企业只有有效整合资源才能有更大的发展，对个人来说，更是如此。个人怎样进行资源整合，最简单的方式就是建立一个系统，也称之为资源圈，团结一群人在一起做一件事情。当今社会是一个互助的社会，帮助很重要，帮助别人有时候就是在帮助自己。想要自己获得帮助，首先给予别人帮助；要想成就自己，就要先想法去成就别人。简单来说就是我为人人、人人为我。

浩为资源堂整合浩为学堂，用户可用积分兑换以下礼品：规则请看前面相关内容，同时为用户提供高校联盟板块组织的实地活动：IT 学习交流、公益活动等。

积分可以通过发表日志、回答别人问题、参与实地活动、提供资源下载等获得，如果你的日志被收录“浩为资源堂月刊”，将获得更多的积分。浩为资源堂已将日志和群组话题合并，你在发表日志可指定群组，自动成为群组话题；你也可指定你的日志查看方式，全力满足你的个性化需要。

浩为资源堂更为积极者提供参与学习小组，管理社区、QQ 群，组织实地活动等机会，致力于提高你的能力。不论是领头人，还是跟随者，成功绝不是你拥有多少物质的及精神上的收益，而是你拥有多少创造持续成功的“资本”。

但从系统论的角度来看：人生不会有所谓的最终的成功，有的只是你对人生的豁达和你看事物的那份坦然。人活的是一份良知和一份责任，活的是一种平凡和一种心态！中国有句古话“盖棺论定”，这也是对中国几千年来成功人士的一个概括和告诫，如果你真想追求那种最终的成功，就脱离不了“盖棺论定”的宿命。浩为资源堂期待更多的人活得精彩。

17.7 聚餐有感：大学生该注意三点

2009 年 12 月初，我老师项目组四个学生请他吃饭，顺便叫上我，因为我曾培训过他们，并且该项目和浩为配置式应用服务器 HoCAS 相关。在《从人生需要引导论 Java 快速开发》的“第 25 章 基于 HoCAS 的 UML 建模”讲述的就是这项目，书中简单介绍了这四个学生：三个大二学生及一个研一学生从年初开始做项目，目前进展不错。期间谈了不少问题，在此总结一下：

一、注意身体。在去饭店的路上，谈到一个同学会晚点过来，因为他感冒了。感冒的一个原因是平时不爱运动，一不小心就着凉了。还说他们连上午班级组织的运动——打羽毛球都没去参加，现在变得很不喜欢活动。身体是革命的本钱，更是自己的本钱，如果在大学期间不注意，等以后发现身体不行了，再去注意就迟了。我还告诉他们，因为第一家单位工作很轻松，就养成了每工作一个小时就休息几分钟的习惯，以致现在工作一般不会觉得很累，建议他们不妨也试试。因为流感挺吓人，我告诉他们可以多按摩按摩虎口上的一个穴位。大学专业为中医的研究生问我，你怎么也知道那个穴位？我说身体是自己的，平时多留心就会减少很多麻烦，这个穴位是我查预防感冒时知道的。

二、珍惜机会。99 年 5 月开始，我在我老师研究所里呆了半年多，开发并实施了好几个项目。因为有这样的工作经历，年底找工作特别容易，并且不差，仅工作一年多，在 02 年初仅靠工资，就在张江按揭了一个 90 平方的房子。说这些只是为了告诉大家，这是因为懂得珍惜机会，如果没有研究所的半年多经历，我找工作估计不会那么轻松。这个项目已经半年多了，有些进展，但没达到预期目标，所以我老师希望我能给他们些帮助（其实一开始就希望我参与，但我时间有限），更希望他们自己多努力。因为该项目采用 UML 建立业务模型，根据业务模型直接生成业务系统，所以实用性很强，也即他们现在做的事情就和以后的工作相差不大。我老师有一句话：“你们很多同学都羡慕你们”，他们为什么会羡慕呢？

三、学会沟通。该项目是那研究生负责进度，三个大学生全力实施。没有达到预期目标，肯定是进度上的问题。为啥会拖后，主要是沟通上的问题。一是实施存在问题时，没能及时解决；而是任务安排下去之后，没能及时跟踪。不跟踪加上沟通不顺利，存在问题就没法在短时间内解决。在实际工作中，很多时间就花在沟通上，你和客户沟通才能了解需求、和上司沟通才能争取支持、和同事沟通才能寻求帮助等等。生活中存在更多需要沟通的地方，比如多和亲人沟通，家庭更加和睦。这在《喻世明言新篇-人生需要引导》中“第 8 章 相处的艺术”中有更多阐述，希望大家理解到沟通的重要性。

当然聚餐谈到的问题远不止这些，但这三个最为重要，然而也只是抛砖引玉，因为更多内容有待浩为教育来为你解答，期望更多的人能领悟这三点，让自己活得更精彩。关注浩为资源堂，你的生活因此而精彩；学以致用，以 IT 技术、职场体验为主，面向大学生的个性化社区式资源圈。

17.8 寄语大学生：社会讲究利益

为了方便《人生需要引导》读者的交流，2009.05.04 开放 QQ 群 78928780 的注册，略加宣传，短短一个半月就变成 4 个群，300 多个有效成员，2009.06.18 统计总数为 361，期间还踢掉好几十人。大部分成员为大学生，作者喻桃阳在写书的同时花了不少时间和他们交流。结合一些同学的调查及交流的结论，归纳出以下几个建议：

1、不少人在浩为 QQ 群不敢发言，说插不上话。面对问题，其实大家都可说说自己的解决办法，不要顾虑太多。你思考了说明自己有进步，能写出来更是进步，如果不去做，永远停留在表面。根据领头人跟随者理论，自己不主动，既做不了跟随者，更做不领头人。希望浩为群能对大家的提高有所帮助，每个人都有自己的困惑，不妨大胆说说，这对自己的成长没有坏处。

2、大多数学生急于见到成果，表现方式如下：

1)、不能立即看到成果的事，不做。因为往往没有耐心做到最后。

2)、认为难度很大的事，不做。因为害怕尝试失败。

3)、当自己认为某事做不成，而别人去做时，就和那人说这件事如何不好、将做不成功，而不会提出自己的建议。

其实这是一个心态问题，在于自己没有恒心及嫉妒别人可能做好。态度决定一切，世上无难事，只怕有心人，要做成一件事，必须经过不懈的努力才行。就如读书，经过 N 年的奋斗才可参加高考，而不是一开始读书就高考。

3、很多学生缺少应有的关心及耐心。

1)、如果所做的事不能很快出成果，他就开始怀疑，甚至放弃。

2)、每个大学生都会说我很关心自己的未来，可没几个能真正做到关心并把握自己的未来，大多只是嘴上说说，却无实际行动。大多边看电影边打游戏，嘴上边喊我将来要怎样怎样。还有一部分人想行动，却不知道该如何行动，该从哪方面努力。

3)、大部分学生很少关注学院的事。除非是学校或学院要求到网站下载一些与自己相关的资料，否则很少去看学校学院新闻。某大学学管科的老师曾对我说，现在很多学生对和自己学业相关的通知都不关心。

听到这老师的话很震惊，但在 QQ 群交流后，觉得也正常。人生需要引导，大学生更需要引导，希望能通过浩为及更多热心人的努力，让他们早日“知道自己要做什么”。

4、很多大学生四年的生活路线：

1)、大一迷茫，但积极参加学校与学院的各项活动，成为社团干事，希望大二能当上学生干部。这一年几乎是在学校社团活动中渡过。

2)、大二寻觅，竞选上学生干部的，继续社团活动，在社团中开始锻炼综合素质；没选上的，有些继续做干事，有些退出。有部分人开始寻找或选择自己的努力方向。

3)、大三分化，找着方向的开始努力或考研；找不着方向的继续找或者放弃，甚至开始游戏大学生活。

4)、大四就业，一部分学生通过各种方式成功就业。另一部分，等待就业。

很多同学在找工作受挫后，选择的是回家等待安排，然后就在宿舍玩电脑、游戏、看电影。根本不去想如何提高自己，让自己有更大的优势去找工作。很多人以为回到县一级比较容易找工作，其实县一级更难找，除非你家关系硬，通过走后门进政府机构等单位；但更多的人只是在等待中浪费自己的青春。

作者建议当你有机会提高自己能力的时候，好好把握机会，争取成为自己命运的主人，而不是命运的仆人。切勿把自己的命运寄托在其他人身，或者根本不去思考自己的将来。

看过一篇文章“社会不等你成长”，文中有一句大实话：社会和学校很不一样。

在校园里，老师不要求学生的物质回报，只要你考试成绩好，人格上没有大毛病，基本上就可以获得老师的欢心，也就是能获得以分数表现的奖励。

社会则很不同，社会更多的是利益的交换，是一个市场。评判标准不再是你的智力优异与否（尽管你的聪明和智慧仍然可以帮助你），而是你能否拿出什么别人想要的东西。这个标准不再由老师决定，而是由众多消费者确定。

千万不要把自己习惯了的校园标准原封不动地带进社会，否则只是“楚才晋不用”。社会讲究利益交换，你可以批评社会短视，但社会不会也没有义务等待你成长与成熟。每一个进入社会的人都是平等，不会考虑你刚毕业，没有经历。只要你失去某次机会，就永远失去了。不象在学校，没及格会给你补考，或者到老师那里求个情改个分数。

所以作者建议在学校的学生不妨好好努力，让自己长点真本事，这只会对自己有好处。

浩为资源堂，你的生活因此而精彩；学以致用，以 IT 技术、职场体验为主，面向大学生的个性化社区式资源圈。浩为学堂，人生需要引导，以多层次 Java 培训为起点，知道自己要做什么：个人定位分析、思维习惯培养、职场体验等，更提供参与有偿项目、领导力锻炼营的机会，详情 howwe.net。陈聚雄从“李开复给大学生的公开信”中找到了前进的方向，加入浩为 QQ 群后，通过努力再成为浩为员工，人生需要引导将为参与者提供更多机会。

17.9 《从人生需要引导论 Java 快速开发》章节说明

有人初看《从人生需要引导论 Java 快速开发》，觉得写得很杂、特别乱，为了减弱这种感觉，特将篇章及 Howwe 系列标题单独列出来。两条线：一条讲技术、一条讲人生，学技术之余不妨看看人生方面的东西；看看人生不妨扫一眼技术，技术其实没人想象的那么难。

有朋友建议我调整篇章，在人生这条线调整成围绕“成功学”来展开，我只笑笑。因为讲成功学的书太多、太滥，再多一本也只是浪费自己的心血而已。“优秀是种习惯”，人生系列力争让读者培养一种良好的习惯。

有人对“代码只是一个过程、编程是一种思想、项目开发更是一个系统工程”，提出“把

生活融入编程，有难度。”我回帖：“编程只是生活的一个表现，思想可以导向良好的生活。从哲学的角度来说，很多事情都是相通的。”对方回帖：“哲学这个词汇很深，归根到底也是思想，用来指导人们思想和行为。代码源于生活，服务于生活。思想正确了，过程只需实践。”我回帖：“看来你已经懂了哲学的实质”。

细读上段，也许能让你更明白篇章安排的必要性。细读之后还是不明白，就把某个系列放到一边，单独看另一个系列。这样还不明白，只能说明该书不适合你，我也不会浪费自己的时间再说什么，放下这本书是你最好的选择。

第 1 篇 入门篇

第 1 章 搭建开发环境

Howwe: 感恩成就人生

第 2 章 激发你的思维潜能

Howwe: 学会留下你的脚印

第 3 章 HELLO WORLD 范例

Howwe: 相处是个过程

第 4 章 数据类型

Howwe: 一生受用不尽的经验

第 5 章 控制结构

Howwe: 睡觉别太晚，器官也有工作表

第 6 章 基于数据库的 HELLO WORLD

Howwe: 销售产品，还是人品

第 7 章 人生没有救世主，一切靠自己

Howwe: 活着是一种责任，多做善事

第 8 章 认识开源数据 HSQLDB

Howwe: 经营人生

第 9 章 SQLEXPLORER 使用简介及 SQL 简介

Howwe: 资源整合

第 10 章 了解 JSP

Howwe: 舍得放弃

第 11 章 JSP+JAVABEAN

Howwe: 对自己的脸负责

第 12 章 数据结构

Howwe: 别做生活的奴隶

第 13 章 JSP+JAVABEAN+SERVLET

Howwe: 生活的真谛

第 14 章 电话外呼系统

Howwe: 服务意识

第 15 章 HWCALL 系统设计

Howwe: 给他人快乐、自己更快乐

第 16 章 HWCALL 编码

Howwe: 我的养生之道

第 17 章 WEB RIA 简介

Howwe: 婚姻：己所不欲，勿施于人

第 18 章 基于 HOCAS 的项目管理系统

Howwe: 男人是婚姻中的孩子，知足的伴侣

第 2 篇 原理篇

第 19 章 WEB 应用工作原理

Howwe: 真能恃才傲物，仅是无知而已

第 20 章 结构、MVC、框架

Howwe: 改变一生的 90/10 原理

第 21 章 软件工程

Howwe: 做人和做事

第 22 章 面向对象

Howwe: 二八定律

第 23 章 SSH 原理

Howwe: 简单生活，学会放弃

第 24 章 HOCAS 表示层原理

Howwe: 人才及用人之道

第 25 章 基于 HOCAS 的 UML 建模

Howwe: 找对人才能做对事，注重责任心

第 26 章 基于 SOA 的 HOCAS

Howwe: 脚踏实地做事

第 27 章 认清 GOOGLE 互联网战略

Howwe: 想成功就要资源整合