

高等职业教育电子信息类专业
“双证课程”培养方案配套教材

国家信息化
计算机教育认证

CEAC

认证教材

程序设计职业核心能力课程 工程化程序设计 (Java)

中国高等职业技术教育研究会 指导
CEAC 信息化培训认证管理办公室 组编



高等教育出版社
Higher Education Press

高等职业教育电子信息类专业“双证课程”培养方案配套教材

程序设计职业核心能力课程

工程化程序设计(Java)

中国高等职业技术教育研究会 指导
CEAC 信息化培训认证管理办公室 组编

高等教育出版社

内容提要

本书采用“就业导向的职业能力系统化课程开发方法(VOCSCUM)”进行开发,是国家教育科学“十五”规划国家级课题“IT领域高职课程结构改革与教材改革的研究与实验”的研究成果之一。本书作为《高等职业教育电子信息类专业“双证课程”培养方案配套教材》之一,同时也是“CEAC国家信息化培训认证”的指定教材,具有鲜明的特色,可作为高职高专院校电子信息类专业教材。

本书是“程序设计”职业能力课程的第三阶段课程的教材,主要涉及程序设计语言Java的应用。Java语言是一种很优秀的语言,具有面向对象、平台无关、安全稳定、可移植性好和多线程等优良特性,是目前软件设计中极为流行的网络编程语言。Java语言不仅可以用来开发大型的应用程序,而且特别适合于Internet的应用开发。本书将面向对象的基本理论与Java语言程序设计相结合,使读者逐渐培养并能正确运用面向对象的思维方法来分析问题和解决问题。

本书共12章,主要包括:面向对象技术的概述及Java的特点,Java基本语法,类与对象,类的封装、继承和多态,接口和包,对象设计的规范,Applet程序,Java的输入与输出流,Java的图形用户界面,Java的多线程,Java数据库开发,Java的Web开发应用等内容。

本书适合于高等职业学校、高等专科学校、成人高等院校、本科院校举办的职业技术学院电子信息类专业教学使用,也可供示范性软件职业技术学院、继续教育学院、民办高校、技能型紧缺人才培养使用。

图书在版编目(CIP)数据

工程化程序设计(Java)/CEAC信息化培训认证管理
办公室组编. —北京:高等教育出版社,2006.1

ISBN 7-04-018421-4

I.工... II.C... III.JAVA语言—程序设计—
高等学校:技术学校—教材 IV.TP312

中国版本图书馆CIP数据核字(2005)第148246号

出版发行 高等教育出版社
社 址 北京市西城区德外大街4号
邮政编码 100011
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京中科印刷有限公司

开 本 787×1092 1/16
印 张 24.5
字 数 590 000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landraco.com>
<http://www.landraco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2006年1月第1版
印 次 2006年1月第1次印刷
定 价 35.40元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 18421-00

高等职业教育电子信息类专业“双证课程”培养方案配套教材

编审委员会

顾	问	张尧学	葛道凯	季金奎	刘志鹏	洪京一
		李宗尧	范唯	吴爱华	宋玲	张方
		尹洪	李维利	周雨阳		
主	任	高林				
委	员	张晓云	杨俊清	姜波	周乐挺	戴荭
		潘学海	王金库	杨士勤	李勤	雷波
课程审定		高林	许远	鲍洁		
内容审定		樊月华	袁枚	王晖	黄心渊	
行业审定		洪京一				
秘书长		曹洪波	杨春慧			

《工程化程序设计(Java)》

主	编	张荣新	
副	主	编	吴卫东 时瑞鹏
参	编	郭景武	孙华峰

国家教育科学“十五”规划国家级课题“IT领域高职
课程结构改革与教材改革的研究与试验”研究成果
高等职业教育电子信息类专业“双证课程”培养方案配套教材

出版说明

目前,我国的高等职业教育正面临着新的形势——以“就业导向、产学结合、推行双证、改革学制、订单培养、打造银领”为主要特点,以培养高技能的技术应用型人才为根本目的。专业建设和课程开发历来是教育改革的核心与突破口。经过十年来的发展,高职教育虽然取得很大进展,但课程模式、教学内容等还有学科系统化的本科压缩型痕迹。尽管从国外引进了许多先进的课程模式和教育思想,但由于国情的不同并且缺少具有中国特色的课程开发方法,目前成功案例也不多。

本套课程改革系列教材采用了经教育部鉴定的“就业导向的职业能力系统化课程及其开发方法”,贯彻了“理念创新、方法创新、特色创新、内容创新”四大原则,在教材建设上进行了改革和探索,是当前高等职业教育教学改革与创新思想的集中体现,主要表现在以下几点:

一、突出行业需求,符合教学管理要求,采用先进开发方法

(1) 依据行业企业需求开发。配套教材是根据信息产业发展对复合型高技能人才需求的特点,并结合最新推出的“CEAC—院校 IT 职业认证证书”标准要求编写而成。认证证书表明持证人具备了相应认证的技术水平和应用能力,它可以作为相关岗位选聘人员、技术水平鉴定的参考依据。将其引入学历教育,可以使高职高专学生在不延长学制的情况下,获得职业证书以提高就业的竞争力。

(2) 依据最新专业目录开发。配套教材以教育部最新制定的《普通高等教育学校高职高专教育指导性专业目录》中的电子信息类大类专业(大类代码:59)设置为依据,进行课程建设。

(3) 采用先进课程开发方法。配套教材采用教育部推荐的“就业导向的职业能力系统化课程及其开发方法(VOCSCUM)”集中反映了高等职业教育课程的基本特征。该方法指出,在高等职业教育突破学科系统化课程模式后,应实施系统化的职业能力课程,在课程模式和开发方法中强调就业导向,产学结合和双证书教育等。VOCSCUM是在高等职业教育课程理论研究的基础上,借鉴国际先进的职业教育课程模式,尤其是澳大利亚和德国的经验,并结合中国国情研制和开发的一套具有自主知识产权的课程模式和开发方法,它适用于两、三年制的高等职业教育。该方法的基本思想已得到教育部领导的肯定,并在教育部组织的高等职业教育四类紧缺人才培养方案制定中进行试用。

二、体现职业核心能力的教材编写思路

上述的思想方法集中体现于《高等职业教育电子信息类专业“双证课程”解决方案(两、三年制适用)》(以下简称“解决方案”)一书中。“解决方案”的出版得到教育部高等教育司、信息产业部信息化推进司、劳动和社会保障部职业技能鉴定中心领导的极大关注和大力支持,并对本书的出版给予了具体的指导。2005年,信息产业部“国家信息化计算机教育认证项目(CEAC)”的管

理机构在“解决方案”的基础上编制了《CEAC 高等职业教育电子信息类专业“双证课程”培养方案》(以下简称“培养方案”),并配套开发了职业认证证书,每个专业的培养方案中,有 7~8 门课程与相应的职业培训证书对应。

根据“培养方案”,我们组织编写了一系列的通识课程教材、职业能力核心课程教材,同时将部分教材作为获得“CEAC—院校 IT 职业认证证书”的认证培训教材。

我们按照 VOCSCUM 课程开发方法的要求,开发纵向为主、横向相关的链路课程(Chain Curriculum)教材,并对程序设计、数据库开发、网络系统配置、网页设计与网站建设、电脑平面设计、电子产品组装与维护等职业核心能力课程中的认证课程,配套研发了立体化教学考核支持系统,以保证这些课程的授课质量。

本系列配套教材不仅覆盖计算机办公应用、软件开发技术、网络技术等常规认证课程,还包括了硬件技术、微电子应用、通信技术、数字制造技术、集成电路设计、应用电子技术、信息管理等专业领域的主要课程,可供高等职业教育电子信息类两、三年制各专业使用。

本系列配套教材将于 2005 年陆续出版,当年先出版 40 余种,其余力争 2006 年底全部完成。

三、不断凝聚、扩大共识,推动高职 IT 课程改革

为了调动广大高等职业学校的优秀教师参加该系列配套教材编写的积极性,相关教材的出版采取“滚动机制”,除了组织示范性链路课程的配套教材出版外,我们还接受有关教师结合自身教学实践并按照“解决方案”编写的教材投稿,经过审核合格后,作为国家教育科学“十五”规划国家级课题——“IT 领域高职课程结构改革与教材改革的研究与试验”的研究成果列入出版计划。热忱欢迎广大高等职业院校电子信息类教师和我们更加深入地研究、引进、摸索、总结 IT 类专业与课程开发经验,通过推广开发的课程,树立高等职业教育品牌,将高等职业教育课程的改革引向深入。

高等职业教育电子信息类专业“双证课程”培养方案配套教材编委会(以下简称:高职电子信息类专业双证配套教材编委会)秘书处设在信息产业部 CEAC 信息化培训认证管理办公室。

本系列配套教材是教育部、信息产业部组织相关专家编写共同推出的双证教材,在信息产业部信息化推进司的领导下,CEAC 信息化培训认证管理办公室专门配套了与课程体系相关的“CEAC—院校 IT 职业认证证书”标准,供高等职业学校在选择 IT 认证培训证书时选用。我们也热忱欢迎其他的职业资格证书和培训证书的管理机构与我们合作,设计出更多的证书体系与课程体系的接口方案。

本系列配套教材是集体的智慧、集体的著作,参加本书编撰工作的人员对社会各界的支持表示感谢。

由于时间仓促,本书不可避免地存在这样或那样的不足,甚至由于学识水平所限,虽竭智尽力,仍难免谬误,希望专家、同行、学者给予批评指正。

高等职业教育电子信息类专业“双证
课程”培养方案配套教材编审委员会
2005 年 8 月

序

我很高兴看到,作为教育部重点课题“高职高专教育课程设置和教学内容体系原则的研究与实践”的研究成果之一,国家教育科学“十五”规划国家级课题——“IT 领域高职课程结构改革与教材改革的研究与试验”课题组所编撰的《高等职业教育电子信息类专业“双证课程”解决方案(两、三年制适用)》(以下简称“解决方案”)以及高等职业教育电子信息类专业“双证课程”培养方案配套教材分别由科学出版社和高等教育出版社出版了。

我国高等职业教育面临着新的转折点。随着国民经济健康、持续的发展,我国越来越需要大批高素质的实用型高级人才。如何培养职业人才呢?教育部提出了“以就业为导向”的指导思想,在这个思想的指导下,高等职业教育的人才培养模式正在发生巨大变革。例如,产学结合、两年学制、推行双证、建设实训基地等,都是围绕就业导向而采取的一系列重要措施。

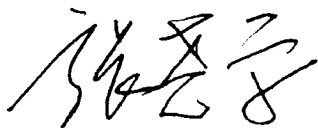
信息产业是我国支柱产业之一,它需要大批高素质的高级实用人才。《高等职业教育电子信息类专业“双证课程”解决方案》以及高等职业教育电子信息类专业“双证课程”培养方案配套教材的出版对促进高等职业教育 IT 类人才培养,我国 IT 产业的发展,进一步改革高等职业教育人才培养模式都具有积极意义,它的创新之处主要在于:

(1)“解决方案”以及配套教材是依据行业企业需求开发的,它根据信息产业发展对复合型高技能人才需求的特点,结合信息产业部最新推出的“CEAC—院校 IT 职业认证证书”标准要求编写而成。认证证书表明持证人具备了相应技术水平和应用能力,它可以作为相关岗位选聘人员、技术水平鉴定的参考依据。将其引入学历教育,可以使高职高专学生在不延长学制的情况下,获得证书以提高就业的竞争力。

(2)“解决方案”以及配套教材是根据教育部最新制定的《普通高等教育学校高职高专教育指导性专业目录》开发的,并以其中的电子信息类专业(大类代码:59)设置的情况为依据,对于高等职业院校两年制 IT 类专业学校来说,具有较大的参考价值。

(3)“解决方案”以及配套教材采取了先进的课程开发方法,采用了已经通过部级鉴定的“就业导向的职业能力系统化课程及其开发方法(VOCSCUM)”。该方法现已作为优秀案例列入教育部高等教育司组织编写的“银领工程”系列丛书,值得高职高专院校借鉴。

我希望,从事 IT 类高等职业教育的老师以及在该领域学习的学生能从“解决方案”以及配套教材中得到较大的收获。



2005 年 6 月 17 日

序

高等职业教育电子信息类专业“双证课程”解决方案和高等职业教育电子信息类“双证课程”培养方案配套教材在课题组成员的努力、众多专家和机构的支持下,终于取得了丰硕的成果。“解决方案”不仅较一年前的初稿有了很大的改进,而且与行业企业的需求越来越近,同时配套教材已由高等教育出版社陆续出版了。

《高等职业教育电子信息类专业“双证课程”解决方案》和高等职业教育电子信息类“双证课程”培养方案配套教材的编撰出版直接源于国家级和教育部的两个课题研究成果。教育部门根据信息产业发展对人才的需求,对高等职业教育的 IT 类课程进行了改革,并大力推进两年制软件职业技术学院的发展。教育课程的改革为行业的发展不断输送适用的技术应用型人才,有力地促进了我国信息化的进程。信息化推进司作为信息产业部负责推进信息化工作的职能部门,积极支持并参与该课题的研究工作,同时责成我司主管“国家信息化计算机教育认证项目”的负责同志为该项目研究提供支持,并配合该项目推出了“CEAC—院校 IT 职业认证证书”标准。

这种由 IT 领域的教育专家和信息产业行业部门合作,在对信息产业行业的人才需求进行调查分析的基础上,有针对性地设计符合信息产业发展需求的人才培养方案,并由行业部门配套职业证书,既有利于培养符合需求、适销对路的人才,又有利于信息产业的发展,也有利于教育部门根据市场需求办学,提高办学效益,这实在是一件双赢的好事。

鉴于“解决方案”配套教材符合“推进信息化建设、促进信息化知识培训”的工作宗旨,我们将支持上述研究成果和教材的推广工作。希望参与该项工作的同志继续努力,以求好上加好、精益求精,为推动信息产业人才培养和我国的信息化建设继续做出更多的贡献。



2005 年 6 月 17 日

前 言

面向对象技术是程序设计方法学的一场革命,已成为计算机应用开发领域的主流技术。面向对象技术能够模拟现实世界的事物,使软件开发人员按照认识客观事物的思维方法进行软件开发。随着网络技术的迅速发展和应用普及,对程序设计语言提出了安全性、移植性好,与平台无关,多线程等更高要求,Java语言正是为满足这些要求而设计的。由于Java语言的平台无关性与因特网发展的紧密结合,它成为目前最流行的软件开发语言之一。Java语言有着广阔的应用前景,它不但适合于网络程序设计,也适合于一般的大规模软件工程项目。有人预言,在不久的将来,全球90%以上的程序代码将是采用Java语言开发的。

本教材是采用获得国家教学成果奖的“就业导向的职业能力系统化课程及其开发方法”(简称VOCSUM)对“计算机软件技术”专业进行职业能力课程的开发成果。根据“计算机软件技术”专业的三大职业核心能力,我们分别建构了“程序设计”、“数据库应用技术”、“软件测试与技术服务”三条职业能力培养的链路课程。每个链路课程反映掌握专门技术从易到难的训练过程,也是理论知识学习的从简到难的过程。

进行课程开发时,我们把每个链路看成一个整体工作任务,从问题中心课程开始,到任务中心课程结束,使学生通过链路课程完整体验实际完成任务的过程。链路课程共分为4个阶段,分别为Step 1、Step 2、Step 3和Step 4。在横向的链路课程形成递进层次关系的同时,纵向的链路课程之间形成相关性。各阶段课程的任务如下:

Step 1: 激发性课程,基于工作过程的技术感受经历。

使学生了解本项工作的整体过程,激发学生学习技术的兴趣,结合工作过程的讲解、技术和相关理论知识的认知做简单介绍,采用问题中心范型的课程。

Step 2: 学科性课程,重点是学科知识的掌握、复用。

使学生掌握本项工作所需要的相关理论知识,部分涉及技术过程,涉及与本职业能力有关的各类学科知识,可以按照学科中心范型的课程或任务中心的课程来组织。

Step 3: 技术性课程,重点是技术知识的掌握、复用。

使学生掌握本项工作所需要的,结合现行具体工作岗位的关键技术技能,同时进一步深化、提高已经学习的理论知识。可以根据国家职业标准、行业技术培训标准,组织培训中心范型的课程。

Step 4: 训练性课程,目的是理论和技术的领会和内化。

通过选取典型的工作过程,编制综合实习、实训课程,全面领会、内化前三个阶段的知识和技

能,同时讲授工作过程中的经验性知识,使得学生成为“高技能人才”预备者。

“程序设计职业能力课程”(链路课程),如下表所示。

表 网络程序设计课程链路

课程阶段	Step1	Step2	Step3	Step4
课程名称	C#	程序设计语言(C)	工程化程序设计(Java)	
课程范型	问题中心	任务中心	培训中心	任务中心
对应职业能力	程序设计能力			
课程基础(起点)				
建议学时		60	128	

本书是软件专业程序设计链路中第三阶段的课程教材,该课程的开发遵循了以应用为目的,从感性认识和实践入手。本书对 Java 语言的内容、功能、特性和实际应用做了深入浅出、系统全面的介绍。全书内容丰富,系统性和应用性强,融入了编者多年的教学 and 实践经验。书中配备了大量的例子,叙述详细且通俗易懂,便于读者自学。另外,每章还附有大量的思考与练习方便读者自测。通过本书的学习,读者能够很快地掌握 Java 语言程序设计的方法和技巧,同时对面向对象的设计也有较深入的理解。

本书的教学大纲分三类指标进行描述:

◇ 理论性目标——应掌握的基本知识、基本理论。

◇ 操作技能性目标——应掌握的基本技能、基本操作。

◇ 经验性目标——在实际应用过程中的实际经验、应掌握的注意事项等。

本书共分 12 章,前 6 章主要介绍面向对象技术和 Java 基本语法,其中包括 Java 基本语法、数据类型、符号和表达式、流程控制、数组和字符串、类和对象、接口、包、异常处理等基础知识。后面 6 章深入地介绍 Java 编程技术,包括 Applet 程序、多线程、Java 的输入/输出流、图形用户界面设计、数据库开发、Java 的 Web 开发应用等内容。

全书由天津职业大学张荣新主编并负责制定编写大纲和统稿工作。本书的具体分工是:第 1、3、4、5、6 章由张荣新编写,第 9、12 章由吴卫东编写,第 8、10、11 章由天津职业大学时瑞鹏编写,第 2 章由铁道第三勘测设计院郭景武编写,第 7 章由天津职业大学孙华峰编写。

在编写过程中,得到了国家教育科学“十五”规划国家级课题组(“IT 领域高职课程结构改革与教材改革的研究与试验”)、CEAC 信息化培训认证管理办公室、高等教育出版社的大力支持和帮助,在此一并表示衷心感谢。

由于计算机网络技术发展迅猛,加之时间仓促,书中难免有不妥或错误之处,希望读者批评指正。

编者

zrx-03@163.com

2005 年 8 月

目 录

第 1 章 概述	1	2.4.2 字符串	51
1.1 面向对象技术概述	1	2.4.3 main() 方法的参数	54
1.1.1 面向对象与面向过程的 比较	2	本章小结	54
1.1.2 面向对象技术的基本要素	4	思考与练习	55
1.2 面向对象程序设计语言 Java	7	第 3 章 类与对象	61
1.2.1 Java 的发展历史	8	3.1 类的概念及定义	61
1.2.2 Java 语言的特点	8	3.1.1 类的定义	61
1.2.3 Java 与 C/C++ 的比较	10	3.1.2 成员变量	63
1.2.4 JDK 的安装、配置与使用	12	3.1.3 方法	65
1.2.5 第一个 Java 程序	13	3.2 对象	70
1.3 集成开发环境 JBuilder 9	16	3.2.1 对象的创建	70
1.3.1 JBuilder 9 的安装和启动	17	3.2.2 对象的使用	71
1.3.2 JBuilder 9 集成开发环境 简介	18	3.2.3 对象的清除	73
1.3.3 JBuilder 下的 Java 编程	22	3.2.4 静态初始化	74
本章小结	24	本章小结	75
思考与练习	24	思考与练习	75
第 2 章 Java 的基本语法	29	第 4 章 类的封装、继承和多态	81
2.1 Java 符号	29	4.1 类的封装	81
2.1.1 注释和关键字	29	4.1.1 对象的可见性和作用域	82
2.1.2 标识符	31	4.1.2 访问权限	83
2.2 Java 语言的基本数据类型、变量 与常量	31	4.2 类的继承性	88
2.2.1 数据类型	31	4.2.1 类继承的实现	88
2.2.2 变量与常量	33	4.2.2 this 和 super 关键字	90
2.2.3 简单数据类型的转换	34	4.2.3 初始化的顺序	94
2.3 运算符、表达式和控制语句	35	4.2.4 对象之间的类型转换	97
2.3.1 运算符与表达式	35	4.3 类的多态性	99
2.3.2 语句	37	4.3.1 方法的重载	99
2.4 数组和字符串	46	4.3.2 方法的覆盖	101
2.4.1 数组	46	4.3.3 final 方法和类	103
		4.3.4 特殊修饰符 native	104
		本章小结	104
		思考与练习	104

第5章 接口和包	112	7.2.1 使用 AWT 绘制图形	160
5.1 抽象类和抽象方法	112	7.2.2 覆盖 init() 方法的 Applet 程序	162
5.1.1 抽象类与抽象方法的声明	112	7.2.3 Applet 各种方法演示	164
5.1.2 抽象类与抽象方法的使用	114	本章小结	167
5.2 内部类	116	思考与练习	167
5.3 接口	117	第8章 Java 的输入与输出流	171
5.3.1 接口的概念	118	8.1 基本输入/输出类	171
5.3.2 接口的声明	118	8.1.1 流的简介	171
5.3.3 接口的使用	120	8.1.2 输入/输出流类	172
5.4 包	122	8.2 输入/输出流类的应用	176
5.4.1 包的声明	122	8.2.1 标准输入	176
5.4.2 包的使用	123	8.2.2 输入/输出流类的典型应用	180
5.5 Java 程序结构	125	8.3 文件操作	185
本章小结	126	8.3.1 建立 File 对象	186
思考与练习	126	8.3.2 File 对象的属性和操作	187
第6章 对象设计的规范	133	本章小结	196
6.1 Object 类及标准包	134	思考与练习	196
6.1.1 Object 类	134	第9章 Java 的图形用户界面	200
6.1.2 java.lang 包	137	9.1 概论	201
6.1.3 Java 的标准包	138	9.1.1 基于组件的开发	201
6.2 Java 的异常处理	139	9.1.2 用户界面设计器	202
6.2.1 异常处理机制	139	9.1.3 组件选项板	203
6.2.2 异常的捕获	141	9.2 AWT 和 Swing	203
6.2.3 抛出异常	143	9.2.1 AWT	203
6.2.4 创建自己的异常	146	9.2.2 Swing	204
6.2.5 异常处理的优点	147	9.2.3 组件	205
6.3 Java 编码的其他规范	148	9.2.4 Swing 组件介绍	205
6.3.1 命名规范	148	9.3 应用程序设计步骤	208
6.3.2 增强程序的可读性	150	9.3.1 创建项目工程	209
本章小结	150	9.3.2 创建应用程序	211
思考与练习	150	9.3.3 调整组件尺寸	215
第7章 Applet 程序	157	9.3.4 增加同一组件的多个副本	215
7.1 Applet 简介	157	9.3.5 检视器属性面板	217
7.1.1 Applet 类	157	9.3.6 属性编辑器	219
7.1.2 Applet 程序的运行过程	159	9.3.7 组件树	220
7.2 Applet 程序举例	160	9.3.8 激活组件设计器	221

9.3.9 容器	222	10.4.1 线程组	292
9.3.10 修改组件名称	224	10.4.2 线程的优先级	294
9.3.11 同时选中多个组件	224	本章小结	295
9.3.12 共享属性	224	思考与练习	295
9.3.13 创建事件处理程序	225	第 11 章 Java 数据库开发	299
9.3.14 组件命名规范	227	11.1 使用 JDBC 连接数据库	299
9.4 布局管理器	228	11.1.1 JDBC 简介	299
9.4.1 概述	228	11.1.2 JDBC 驱动程序	299
9.4.2 容器与布局管理器	229	11.2 建立数据源	300
9.4.3 布局管理器综合实例	239	11.2.1 建立数据库	300
9.5 菜单设计器	245	11.2.2 建立数据源	303
9.5.1 概述	245	11.3 JDBC 数据库开发基础	306
9.5.2 组件选项板上的菜单组件	245	11.3.1 数据库开发中涉及的	
9.5.3 在 JBuilder 中如何使用菜单		几个类	306
设计器	248	11.3.2 数据库编程的一般过程	308
9.6 对话框的设计	260	11.4 数据库编程实例	309
9.6.1 JoptionPane 对话框	260	11.4.1 查询数据库	309
9.6.2 Dialog 对话框	265	11.4.2 插入记录	311
9.7 键盘和鼠标事件处理	266	11.4.3 修改记录	312
9.7.1 键盘事件	266	11.4.4 删除记录	313
9.7.2 鼠标事件	268	11.4.5 建立表	314
本章小结	272	11.4.6 取表中各列名称	315
思考与练习	273	11.4.7 删除表	317
第 10 章 Java 的多线程	278	本章小结	317
10.1 多线程的概念	278	思考与练习	318
10.1.1 线程的生存周期	279	第 12 章 Java 的 Web 开发应用	322
10.1.2 创建线程	279	12.1 Java Servlet 程序开发	323
10.2 控制线程	280	12.1.1 Servlet 简介	323
10.2.1 启动线程	281	12.1.2 Servlet 的基本方法	326
10.2.2 使线程睡眠	281	12.1.3 程序实例	329
10.2.3 中断线程	282	12.2 JavaBean	332
10.2.4 停止线程	284	12.2.1 JavaBean 简介	332
10.3 互斥线程间的同步机制	285	12.2.2 JavaBean 在 JSP 中的	
10.3.1 多线程带来的问题	285	应用	343
10.3.2 临界区	286	12.3 Web 开发实例	353
10.3.3 线程等待	288	本章小结	370
10.4 线程管理	291	思考与练习	370
		参考文献	375

第 1 章 概 述



要求

- 了解面向对象技术概念及对 Java 的初步认识



知识点

- 了解面向过程与面向对象技术的区别
- 了解 Java 的发展历史
- 理解面向对象技术的概念及基本要素
- 理解 Java 的特点及与 C/C++ 的区别
- 理解 Java 应用程序和小程序的结构及源文件命名的要求



技能点

- 熟练掌握对 JDK 的环境进行配置及使用
- 熟练掌握使用集成开发环境 JBuilder 9



重点和难点

- 由面向过程向面向对象技术观念的转变
- Java 的特点及与 C/C++ 的区别
- Java 编程工具 JDK 的环境配置及使用和集成开发环境 JBuilder 的使用

1.1 面向对象技术概述

随着计算机技术的不断发展,其软、硬件之间的差距越来越大,造成了计算机发展的不均衡。当系统较为复杂时,常规的软件工具、技术和概念已不足以应付,从而使软件开发陷入了困境,即所谓的“软件危机”。软、硬件发展的这种差距自计算机出现以来始终存在,但 20 世纪 90 年代后这种差距变得更加明显,在这一背景下,面向对象技术逐步兴起。随着面向对象技术的不断完善并逐步进入实用阶段,它受到广大软件开发者的青睐,吸引了众多的人士去研究和使用的,使之在软件工程、人工智能等领域得到了十分广泛的应用。人们把面向对象的程序设计语言看做是计算机软件产业的一次革命。

面向对象技术是一种全新的设计和构造软件的技术,它使计算机解决问题的方式更符合人类的思维方式,更能直接地描述客观世界。它通过增加代码的可重用性、可扩充性和程序自动生成功能来提高编程效率,并且大大减少软件维护的开销。现在,面向对象技术已经被越来越多的软件设计人员所接受。

1.1.1 面向对象与面向过程的比较

1. 概念比较

在面向对象程序设计(Object Oriented Programming, OOP)方法出现之前,程序员用面向过程的方法开发程序。人们为了更轻松地解决问题,总是希望尽可能抽象地描述问题、解决问题,过于具体有时候会使人们感到很痛苦。在面向过程中,我们的方法是“先做什么,再做什么,最后做什么”,这就好像一个成年人做事的方法,他知道每一步该干什么,他按部就班地去做。在面向对象中,我们的方法是“不管对象是怎么实现的,我们只需知道它可以干什么,怎么让它去干”,这就像一个孩子的做事方法,他不知道那个东西是怎么来的,他只知道那个东西可以用来干什么以及利用它去干他想干的事。

面向对象技术是一种以对象为基础,以事件或消息来驱动对象执行处理的程序设计技术。它以数据为中心而不是以功能为中心来描述系统,数据相对于功能而言具有更强的稳定性。它将数据和对数据的操作封装在一起,作为一个整体来处理,采用数据抽象和信息隐蔽技术,将这个整体抽象成一种新的数据类型——类,并且考虑不同类之间的联系和类的重用性。类的集成度越高,就越适合大型应用程序的开发。另一方面,面向对象程序的控制流程由运行时各种事件的实际发生来触发,而不再由预定顺序来决定,更符合实际需要。事件驱动程序的执行围绕着消息的产生与处理,靠消息循环机制来实现。更重要的是,利用不断扩充的框架产品 MFC(Microsoft Foundation Classes),在实际编程时可以采用搭积木的方式来组织程序,站在“巨人”肩上实现自己的愿望。面向对象的程序设计方法使得程序结构清晰、简单,提高了代码的重用性,有效地减少了程序的维护工作量,提高了软件的开发效率。

例如,用面向对象技术来解决员工管理方面的问题。重点应该放在员工身上,要了解在管理工作中,员工的主要属性,要对员工做些什么操作等,并且把它们作为一个整体来对待,形成一个类,称为员工类。可以建立许多具体的员工作为其实例,而每一个具体的员工就是员工类的一个对象。员工类中的数据和操作可以提供给相应的应用程序共享,还可以在员工类的基础上派生出经理类、工人类或工程师类等,实现代码的高度重用。

2. 程序设计方法比较

(1) 面向过程程序设计方法

程序由很多模块构成,这些模块可以被单独设计、编码和测试,然后被组装成一个完整的程序。在面向过程的编程语言中,比如 C 或者 Pascal,这些模块就是过程。一个过程就是一个语句序列。例如在下面的程序段中,这个语句序列就是一个过程。

```
int find_max(int num1, int num2)
{
    if ( num1 > num2 )
        return num1;
    else
        return num2;
}
```

像这样的传统编程语言的过程都是由赋值语句、分支、循环和子过程调用这样的命令式语句序列构成,所有的过程都被称为函数(Function),它把参数映射为一个返回值。上面这个函数有两个整数参数 num1 和 num2,并返回一个整数值。一个 if - else 语句分支判断出两个参数中较大的一个,并将它返回。在面向对象的语言中,例如 Java,像上面这样的函数都被称为方法(Method)。

面向过程这种设计方法注重于以下几点:

- ◇ 从程序设计范式上看,程序 = (过程)定义 + (过程)调用。
- ◇ 从程序语言角度出发,在这种方法中,依靠各种循环、分支、函数调用、参数传递来维护程序。
- ◇ 这种方法重视程序流程,不在乎数据的传递方式,只要程序能运行就可以了。

其缺点是:

- ◇ 过程的抽象级别太低,难以重用。
- ◇ 需要花费相当高的成本维护程序与程序之间的关联性。
- ◇ 数据和操作相分离,当一个或多个数据的结构发生变化时,这种变化将波及程序的很多部分甚至遍及整个程序。往往是变量和参数传来传去,传到最后甚至不知道这个变量的作用何在,所以程序的复杂性很难控制。

(2) 面向对象程序设计方法

在面向对象程序设计方法中,客观世界中的问题被看做客观世界中实体与实体之间的关系及其相互作用。每一个实体就是一个对象(Object),其中包含有关数据和一组对这些数据的操作。也就是说,面向对象程序的基本元素为对象。程序结构一般由类的定义和类的使用两部分组成。程序中的一切操作都是通过向对象发送相应的消息来实现,对象接收到消息后,启动有关的方法完成相应的操作。

面向对象这种设计方法注重于以下几点:

- ◇ 从面向对象程序设计范式上看:程序 = 对象 + 消息。
- ◇ 直观性。对象是集数据与操作于一体的,以这种方式进行分析与设计完全符合人类的直觉,能对现实世界的分类系统进行自然的描述。
- ◇ 提高程序的可复用性。
- ◇ 改善程序的可维护性,减轻维护负担。面向对象的封装性,使得维护软件更加容易。只要类提供的方法(接口)不变,方法体的任何修改都不会导致程序的修改。
- ◇ 控制程序的复杂性。可以从设计完好、可复用的对象中构建出复杂的软件,设计良好的、面向对象的软件可以让用户迅速对环境中的变化做出反应。

其缺点是:

- ◇ 面向对象需要完全改变用户的思想。由于长期受到面向过程的影响,这种影响是在潜移默化中形成的,不要低估改变思想的难度,这要求用户先在培训、教育和工具上投资,以达到从理念上对面向对象的认识,这需要一个长期的过程。
- ◇ 面向对象需要更关注于需求、分析和设计。

总之,面向对象的方法与面向过程的方法最大的不同在于,它是以数据为中心而不是以功能为中心,将数据及相应操作封装在一起,抽象成一种新的数据类型——类。另外,面向过程的方法

法是一句接一句地编写程序,面向对象的方法主要是适当地创建对象、修改对象属性和编写对象的方法程序。另外,面向对象程序的控制流程由运行时各种事件的实际发生来触发,而不再由事件的预定顺序来决定。事件驱动程序的执行围绕着消息的产生与处理,靠消息循环机制来实现。

1.1.2 面向对象技术的基本要素

1. 面向对象技术中的几个重要概念

类与对象是面向对象程序设计中 most 基本、最重要的两个概念,有必要仔细理解和彻底掌握,它们将贯穿全书并且逐步深化。

(1) 对象与类

与人们认识客观世界的规律一样,面向对象技术认为客观世界是由各种各样的对象组成,每种对象都有各自的内部状态和运动规律,不同对象间的相互作用和联系就构成了各种不同的系统和客观世界。在面向对象的程序中,客观世界被描绘成一系列完全自治、封装的对象,这些对象通过外部接口访问其他对象。可见,对象是组成一个系统的基本逻辑单元,是一个有组织形式、含有信息的实体。而类是创建对象的样板,在整体上代表一组对象,设计类而不是设计对象可以避免重复编码,类只需要编码一次,就可以创建本类的所有对象。

对象(Object)是客观物体或概念性东西的抽象描述。每个对象都包含本身的属性(Attribute),存在于一定的状态,并有内在的标识。此外,还可以给对象定义一组操作(Operation),用以访问、提取或改变其状态。对象通过其操作给外界展示某些特征,这就是对象的行为(Behavior),所以有时又称对象是由属性和行为两部分组成。对象是包含客观事物特征的抽象实体,是属性和行为的封装体。在程序设计领域,可以用“对象=数据+作用于这些数据上的操作”这一公式来表达。

类(Class)是具有相同属性和行为的一组对象的集合,它为属于该类的全部对象提供了统一的抽象描述,其内部包括属性和行为两个主要部分,类是对象集合的再抽象。

类与对象的关系如同一个模具与用这个模具铸造出来的铸件之间的关系。类给出了属于该类的全部对象的抽象定义,而对象则是符合这种定义的一个实体。所以,一个对象又称作类的一个实例(Instance)。

在面向对象程序设计中,类的确定与划分非常重要,是软件开发中关键的一步,划分的结果直接影响到软件系统的质量。如果划分得当,既有利于程序进行扩充,又可以提高代码的可重用性。因此,在解决实际问题时,需要正确地进行分“类”。理解一个类究竟表示哪一组对象,如何把实际问题中的事物汇聚成一个个的“类”,而不是一组数据,这是面向对象程序设计中的一个难点。

(2) 消息和事件

消息(Message)是描述事件发生的信息,事件(Event)由多个消息组成。消息是对象之间发出的行为请求。封装使对象成为一个相对独立的实体,而消息机制为它们提供了一个相互间动态联系的途径,使它们的行为能互相配合,构成一个有机的运行系统。

程序的执行取决于事件发生的顺序,由顺序产生的消息驱动,而不必预先确定消息产生的顺序,更符合客观世界的实际。