

Java CHENGXU SHEJI ANLI JIAOCHENG

Java程序设计 案例教程

Java程序设计案例教程

主编 李海涛
陈涛
孙红丽

ISBN 978-7-5608-9099-9



定价: 59.80元

同济大学出版社
TONGJI UNIVERSITY PRESS

Java CHENGXU SHEJI ANLI JIAOCHENG

Java程序设计 案例教程

主编 李海涛 陈涛 孙红丽

同济大学出版社
TONGJI UNIVERSITY PRESS



Java CHENGXU SHEJI ANLI JIAOCNENG

Java 程序设计 案例教程

主 编 李海涛 陈 涛 孙红丽
副主编 李玉萍 刘冠秀 张 菁
朱 琳 朱 民

同济大学出版社·上海

内 容 提 要

本书是在“基于翻转课堂的课程教学范式改革”与校本应用型教材建设项目的基礎上编写的。本书采用案例驱动的方式编写,即把理论知识点的讲解融入实际的案例中,通过案例教学的方式达到理论与实践融合,旨在培养学生的动手实践和程序设计能力。全书内容从Java SE基础知识开始,具体包括Java开发环境搭建、Java语言基础、面向对象编程、数组、集合、异常处理、图形界面设计等内容,知识安排从易到难,逐步深入。

本书适合作为高等院校Java程序设计课程的教材,也可供广大技术人员参考。

图书在版编目(CIP)数据

Java 程序设计案例教程 / 李海涛, 陈涛, 孙红丽主
编. --上海: 同济大学出版社, 2021. 6
ISBN 978 - 7 - 5608 - 9099 - 9

I. ①J… II. ①李… ②陈… ③孙… III. ①JAVA 语
言 - 程序设计 - 高等学校 - 教材 IV. ①TP312. 8

中国版本图书馆 CIP 数据核字 (2021) 第 105847 号

Java 程序设计案例教程

李海涛 陈 涛 孙红丽 主编

责任编辑 张平官 责任校对 徐春莲 封面设计 刘文东

出版发行 同济大学出版社 www.tongjipress.com.cn
(地址: 上海市四平路 1239 号 邮编: 200092 电话: 021 - 65985622)

经 销 全国各地新华书店
印 刷 大厂回族自治县聚鑫印刷有限责任公司
开 本 850 mm×1 168 mm 1/16
印 张 19.5 插页 1
字 数 499 000
版 次 2021 年 6 月第 1 版
印 次 2024 年 1 月第 2 次印刷
书 号 ISBN 978 - 7 - 5608 - 9099 - 9

定 价 59.80 元

本书若有印装质量问题, 请向本社发行部调换 版权所有 侵权必究

随着信息技术的发展,传统的教育模式已难以满足学生学习与就业的需要。因此,积极推进教学范式和内容的改革,打破传统的以理论知识传授为主的教学方式,转向注重实践应用与就业能力的培养方式,已成为大多数高等院校的共识。

本书是在“基于翻转课堂的课程教学范式改革”与校本应用型教材建设项目的基礎上编写的。从 Java SE 基础知识开始介绍,具体包括 Java 开发环境搭建、Java 语言基础、面向对象编程、数组、集合、异常处理、图形界面设计等内容,知识安排从易到难,逐步深入,使学生能够掌握 Java 编程技术的基础知识,具备理论联系实际的能力;强化学生的编程思想和编程能力;增加学生的团队合作意识;促进学生职业能力发展,为其以后走向社会或继续深造打下基础。

本书采用案例驱动的方式编写,即把理论知识点的讲解融入实际的案例中,通过案例教学的方式达到理论与实践的融合。例如,在循环结构的学习中通过讲解求多项式的和、求 100 以内的所有素数和猜数字游戏等案例,介绍 while 循环和 for 循环的语法结构、执行流程及二者之间的不同。在数组部分的内容中引入扫雷游戏的随机布雷、扑克牌的随机发牌等案例来激发学生的学习兴趣并提高其创新实践能力。书中的每个案例均附有微视频,学生可以通过扫描二维码观看学习视频进行预习或巩固。

本书主要内容及参考学时如下表:

章 序	主要内容	学 时
第 0 章	从总体上对企业 HR 管理系统进行介绍,具体包括功能模块、数据库设计、界面展示和项目创建等方面	3
第 1 章	本章介绍 Java 语言的来历、Java 语言的特点、Java 虚拟机的概念和作用及面向对象程序设计的相关知识	
第 2 章	本章介绍 Java 语言开发环境的基础知识。先介绍 Java 语言开发环境 JDK 的安装和设置、几种 Java 集成开发环境,再介绍 Java Application(Java 应用程序)与应用在 WWW 上的 Java Applet(Java 小应用程序)的区别	3
第 3 章	本章介绍 Java 语言的基础知识。首先介绍 Java 语言的基本语法、几种不同的数据类型、常量与变量的定义和使用、一些常见运算符的使用;然后介绍三种流程控制语句;最后介绍方法的定义和重载	10
第 4 章	本章介绍面向对象程序设计的核心内容,即类的定义与对象的使用。首先介绍类的定义及使用方法、对象的创建及使用方法;然后介绍访问控制符的使用、构造方法的使用;最后介绍关键字 static 与 final 的使用	4

续表

章 序	主要内容	学 时
第 5 章	本章介绍面向对象三大特性,即封装、继承及多态。首先介绍封装的概念及其用法;然后介绍继承、继承中方法的覆盖与重载以及抽象类和接口的实现;最后介绍多态、抽象类和抽象方法、接口以及 Java 的访问权限	8
第 6 章	本章介绍 Java 语言中一维数组、二维数组的创建及使用方法。首先介绍一维数组的定义、声明、初始化及使用方法;然后介绍二维数组的声明、初始化及使用方法;最后简要介绍多维数组	4
第 7 章	本章首先介绍包的概念、包的应用以及包的定义与导入;然后介绍系统类库中几种常用的类	4
第 8 章	本章介绍 Java 语言中的异常和多线程。首先介绍异常的概念、分类及两种不同处理方式的异常类型,以及处理异常的两种不同方式;然后介绍 Java 语言实现多线程的两种方式,线程生命周期中几种不同的状态及其相互转换、调度方式,以及实现线程同步的方法	6
第 9 章	本章介绍 Java 输入输出的有关概念和常用的类。首先介绍输入流和输出流的概念,然后介绍几种常用的输入流和输出流的常用操作,最后介绍 File 类对文件的操作方法和 RandomAccessFile 类对文件随机读写的方法	6
第 10 章	本章首先介绍图形用户界面概述,然后介绍常用事件和事件处理机制、布局管理,最后介绍 Swing 组件	6
第 11 章	本章首先介绍 SQL 语句和 JDBC 数据库编程基础知识,以及对数据库进行增、删、改、查操作的方法;然后介绍事务管理和数据库连接池,并通过综合案例,结合之前所学知识开发、完善企业 HR 管理系统	8

本书以案例为导向,在结合大量案例及教学经验的基础上进行理论讲解,从第 1 章开始,每章都按照“学习目标—理论讲解—提出任务—任务实施—本章小结—课后练习”的顺序进行编写,每个知识点后都有相关的案例支撑,包括实际案例、运行结果和案例解析,深入浅出地介绍了利用面向对象编程的内容和方法,重实践而不轻理论。

本书由李海涛、陈涛、孙红丽担任主编,李玉萍、刘冠秀、张菁、朱琳和朱民担任副主编,同时邀请校外企业专家共同参与编写。其中,李海涛编写第 0 章,李玉萍编写第 1 章和第 2 章,刘冠秀编写第 3 章,孙红丽编写第 4 章和第 6 章,朱琳编写第 5 章,朱民编写第 7 章和第 10 章,陈涛编写第 8 章和第 9 章,张菁编写第 11 章。

编者虽然倾注了许多精力,但由于水平有限,书中难免存在疏漏,敬请读者批评指正。

编 者

目 录

Contents

第 0 章 企业 HR 管理系统项目介绍	1
学习目标	1
0.1 认识企业 HR 管理系统	1
0.1.1 功能模块	1
0.1.2 系统功能模块图	2
0.2 任务 1:设计并创建企业 HR 管理系统数据库	2
0.2.1 设计数据库表	3
0.2.2 编写数据库脚本	4
0.2.3 导入数据库	6
0.3 任务 2:体验企业 HR 管理系统的功能	7
0.3.1 登录界面	7
0.3.2 用户注册界面	7
0.3.3 用户角色界面	8
0.3.4 汇报工作界面	9
0.3.5 评测员工成绩界面	10
0.3.6 更改角色	11
0.3.7 修改密码	12
0.3.8 查看个人简历	12
0.3.9 系统使用说明	13
本章小结	15
第 1 章 Java 开发入门	16
学习目标	16
1.1 Java 的来历	16
1.2 Java 语言的特点	17
1.3 Java 虚拟机	18
1.4 面向对象程序设计	18
1.4.1 抽象	18
1.4.2 封装	19

1.4.3 类与对象的概念	19
1.5 任务:创建企业 HR 管理系统中的类和对象	20
本章小结	26
课后练习	26
第 2 章 Java 程序开发基础	27
学习目标	27
2.1 Java 开发环境的搭建	27
2.1.1 JDK 的三种版本	27
2.1.2 JDK 的安装	27
2.1.3 JDK 目录介绍	28
2.1.4 环境变量的配置	29
2.1.5 验证配置环境变量	30
2.2 Java 集成开发环境简介	31
2.2.1 Eclipse	31
2.2.2 EditPlus	33
2.3 两类“Hello World”程序	33
2.3.1 Java Application	33
2.3.2 Java Applet 程序	35
2.3.3 两类程序的区别	36
2.4 Java Application 程序的输入/输出	37
2.4.1 字符界面的输入/输出	37
2.4.2 图形界面的输入/输出	37
2.5 任务:利用 Eclipse 创建系统项目	37
本章小结	39
课后练习	39
第 3 章 Java 语法基础	40
学习目标	40
3.1 Java 程序的基本组成	40
3.1.1 标识符	41
3.1.2 关键字	41
3.2 数据类型	41
3.2.1 整数类型	42
3.2.2 浮点类型	43
3.2.3 字符类型	44
3.2.4 布尔类型	44
3.2.5 数据类型之间的相互转换	44

3.3	常量与变量	47
3.3.1	常量	47
3.3.2	变量	49
3.4	运算符	52
3.4.1	赋值运算符	52
3.4.2	算术运算符	53
3.4.3	关系运算符	55
3.4.4	逻辑运算符	56
3.4.5	位运算符	58
3.4.6	运算符的优先级	61
3.5	流程控制语句	61
3.5.1	顺序结构	62
3.5.2	选择结构	62
3.5.3	循环结构	72
3.6	方法	79
3.6.1	方法的定义	79
3.6.2	方法的重载	82
3.7	任务 1:根据用户输入的菜单选项显示与其对应的界面内容	84
3.8	任务 2:显示工作汇报列表	86
	本章小结	87
	课后练习	87
第 4 章	类与对象	89
	学习目标	89
4.1	类的定义	89
4.1.1	类的声明	91
4.1.2	成员变量和常量	92
4.1.3	方法的声明与实现	94
4.2	对象的创建	96
4.2.1	对象的创建	96
4.2.2	对象的使用	96
4.3	构造方法的定义与重载	99
4.4	static 关键字	102
4.4.1	静态变量	103
4.4.2	static 方法	104
4.5	final	105
4.5.1	final 域	105
4.5.2	final 方法	106

4.5.3 final 类	107
4.6 任务:用户管理功能实现	107
本章小结	109
课后练习	109
第5章 面向对象三大特性及其 Java 实现	111
学习目标	111
5.1 封装	111
5.1.1 封装的概念	111
5.1.2 封装的实现	111
5.2 继承	113
5.2.1 Java 的继承	113
5.2.2 方法的覆盖与重载	117
5.2.3 this 与 super 关键字	119
5.3 多态	124
5.3.1 多态的实现	124
5.3.2 多态中的问题	127
5.4 抽象类和抽象方法	128
5.4.1 抽象类	128
5.4.2 抽象方法	129
5.5 接口	130
5.5.1 接口的定义	130
5.5.2 接口的实现	131
5.6 Java 的访问权限	132
5.7 任务 1:显示不同用户的功能菜单	133
5.8 任务 2:处理各角色工作任务及薪资	139
本章小结	145
课后练习	145
第6章 数组	146
学习目标	146
6.1 一维数组	146
6.1.1 一维数组的声明	146
6.1.2 一维数组的初始化	146
6.1.3 一维数组的使用	147
6.2 二维数组	148
6.2.1 二维数组的声明	148
6.2.2 二维数组的初始化	149
6.2.3 二维数组的使用	149

6.3 任务:应用数组实现信息存储	151
本章小结	153
课后练习	153
第7章 Java 类库基础	155
学习目标	155
7.1 包	155
7.1.1 包的概念	155
7.1.2 包的应用	156
7.2 常用的基础类库	159
7.2.1 Object 类	159
7.2.2 数据类型类	161
7.2.3 System 类与 Class 类	163
7.2.4 Math 与 Random	169
7.2.5 String、StringBuilder 和 StringBuffer	172
7.3 任务:利用包对类进行分类管理及常用类的使用	182
本章小结	184
课后练习	184
第8章 异常与多线程	186
学习目标	186
8.1 异常	186
8.1.1 异常与异常类	186
8.1.2 抛出异常	189
8.2 异常的处理	190
8.2.1 捕获异常	190
8.2.2 继续向上抛出异常	192
8.3 多线程	194
8.3.1 Java 多线程的实现	194
8.3.2 线程的生命周期	197
8.3.3 线程的调度	198
8.3.4 线程同步	200
8.4 任务:处理输入菜单选项输入异常	205
本章小结	211
课后练习	212
第9章 Java I/O 系统	213
学习目标	213
9.1 输入输出流概述	213

9.1.1 字节流	213
9.1.2 字符流	215
9.2 几种字节输入流	215
9.2.1 字节数组输入流 ByteArrayInputStream	215
9.2.2 文件输入流 FileInputStream	216
9.3 几种常用的字符输入流	217
9.3.1 字符数组输入流 CharArrayReader	217
9.3.2 字符串输入流 StringReader	218
9.4 几种字节输出流	219
9.4.1 字节数组输出流 ByteArrayOutputStream	219
9.4.2 文件字节输出流 FileOutputStream	220
9.5 文件类 File 和随机存取文件类 RandomAccessFile	221
9.5.1 File 类	221
9.5.2 RandomAccessFile 类	223
9.6 任务:利用输入输出流保存和读取数据	227
本章小结	230
课后练习	230
第 10 章 图形用户界面	232
学习目标	232
10.1 图形用户界面概述	232
10.1.1 AWT 与 Swing	232
10.1.2 AWT 常用组件	232
10.2 事件处理	237
10.2.1 事件处理机制	237
10.2.2 常用事件的分类	239
10.2.3 事件适配器	244
10.3 布局管理	245
10.3.1 边框布局管理器	245
10.3.2 流式布局管理器	247
10.3.3 卡片布局管理器	248
10.3.4 网格布局管理器	250
10.3.5 网格包布局管理器	251
10.3.6 盒式布局管理器	251
10.4 Swing 组件	253
10.4.1 JFrame 窗体	253
10.4.2 JDialog 对话框	254
10.4.3 JPanel 面板	254

10.4.4 JLabel 标签	254
10.4.5 JButton 按钮	255
10.4.6 JTextField 单行文本框	255
10.4.7 JPasswordField 密码框	257
10.4.8 JTextArea 文本域	258
10.4.9 菜单	258
10.5 任务:使用图形用户界面方式实现企业 HR 管理系统	265
本章小结	277
课后练习	277
第 11 章 JDBC 数据库编程	278
学习目标	278
11.1 数据库概述	278
11.1.1 数据库基本概述	278
11.1.2 SQL 简介	278
11.2 JDBC 概述	280
11.3 JDBC 访问数据库	280
11.3.1 使用 JDBC 连接数据库	280
11.3.2 操作数据库——增、删、改、查	284
11.4 JDBC 进阶	293
11.4.1 事务管理	293
11.4.2 数据库连接池	295
11.5 任务:使用 JDBC 连接数据库进行操作	297
本章小结	300
课后练习	300
参考文献	301

企业 HR 管理系统项目介绍

学习目标

了解企业 HR 管理系统功能模块说明,企业 HR 管理系统数据库系统设计,企业 HR 管理系统功能模块设计;会配置企业 HR 管理系统;能熟练操作企业 HR 管理系统。

本章将向读者详细介绍本书所用的案例系统——企业 HR 管理系统的基本设计。企业 HR 管理系统的分析和设计主要包括系统需求分析、系统功能模块设计、系统界面设计、系统主要流程设计、数据库设计和系统详细设计。

0.1 认识企业 HR 管理系统

企业 HR 管理系统是一个结合 Java 语言基础语法知识实现的入门案例,包含类和对象、继承、抽象类和接口、异常、流、Swing 和 JDBC 等知识,主要采用 Eclipse 集成开发环境(integrated development environment,IDE)和 MySQL 数据库。该系统要求实现用户的登录、注册、角色管理、汇报、评测等功能,数据存储在数据库中。

0.1.1 功能模块

1. 用户注册/登录

系统考虑到企业用户管理的真实性,在使用之前需要进行已有账户登录,如果不存在,可以通过注册的方式登记相关基本信息成为系统注册用户,注册用户登录系统后可以相应操作。

2. 角色划分

系统共划分为三个角色,即 staff(普通员工)、manager(经理)和 admin(管理员),可以根据不同角色显示不同的功能菜单。

3. 汇报工作

为了模拟企业真实情况,系统设计普通员工每天需要向部门经理发送工作汇报,经理可以查看员工汇报的工作。

4. 评测员工成绩

企业内的经理每隔一段时间需要对所在部门的员工进行成绩评定,体现员工的工作状态。

普通员工可以在部门经理评测后查看评测的成绩。

5. 个人设置

- (1)查看个人信息。登录的用户可以查看个人基本简历信息。
- (2)注销。员工在操作结束后,可以注销自己的账号,以保障账号的安全。

0.1.2 系统功能模块图

根据企业 HR 管理系统的业务需求、功能需求和用户需求信息绘制的企业 HR 管理系统功能模块如图 0-1 所示,它详细描述了企业 HR 管理系统所具有的功能。

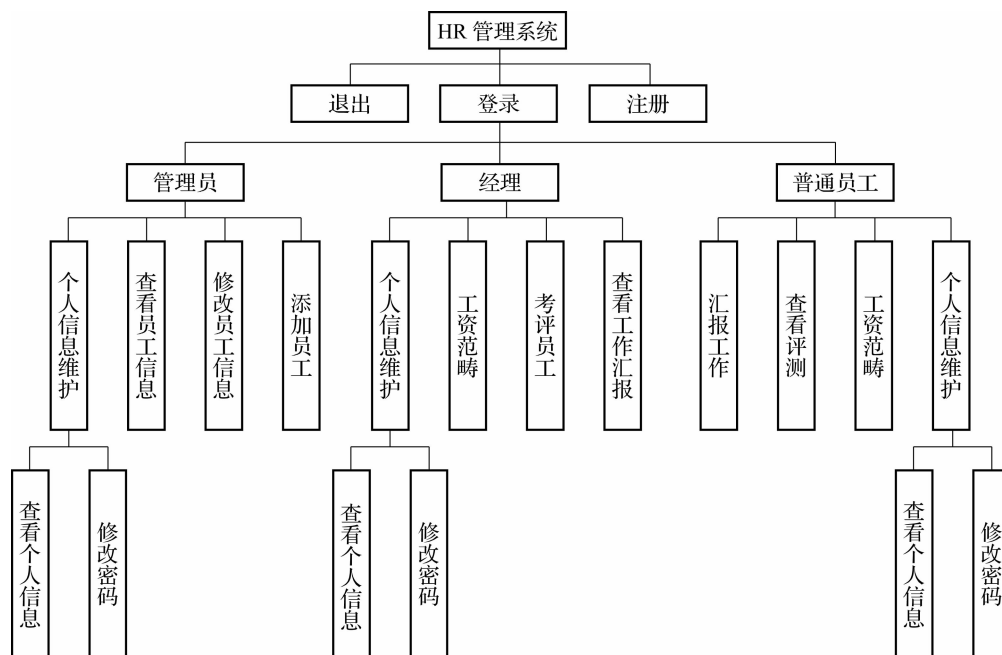


图 0-1 企业 HR 管理系统功能模块

0.2 任务 1:设计并创建企业 HR 管理系统数据库

【任务目标】

设计并创建企业 HR 管理系统的后台数据库。

【任务要点】

打开 MySQL 管理工具,执行本书所附的企业 HR 管理系统的 SQL 脚本创建企业 HR 管理系统数据库。

【任务实施】

- (1)设计数据库、表和视图等对象。
- (2)编写 SQL 脚本。
- (3)选择 MySQL 管理工具创建数据库。

下面对任务实施过程进行详细讲解。

0.2.1 设计数据库表

根据系统功能描述和实习业务分析,进行企业 HR 管理系统的设计,主要数据表内容如下。

1. hr_user(用户信息表)

用户信息表的详细信息见表 0-1。

表 0-1 hr_user

表序号	1	表名		hr_user		
含义	存储用户的基本信息					
序号	属性名称	含义	数据类型	长度	为空性	约束
1	userid	用户 ID	int	11	not null	主键
2	empno	员工编号	varchar	20	not null	
3	username	用户名	varchar	50	not null	
4	password	密码	varchar	100	not null	
5	roleid	角色 ID	int	11	not null	外键
6	departid	部门 ID	int	11	not null	外键
7	salary	薪资	float	8	null	

2. hr_department(部门表)

部门表的详细信息见表 0-2。

表 0-2 hr_department

表序号	2	表名		hr_department		
含义	存储部门的基本信息					
序号	属性名称	含义	数据类型	长度	为空性	约束
1	departid	部门 ID	int	11	not null	主键
2	name	部门名称	varchar	255	not null	

3. hr_role(用户角色表)

用户角色表的详细信息见表 0-3。

表 0-3 hr_role

表序号	3	表名		hr_role		
含义	存储用户角色的基本信息					
序号	属性名称	含义	数据类型	长度	为空性	约束
1	roleid	角色 ID	int	11	not null	主键
2	rolename	角色名称	varchar	255	not null	

4. hr_evaluation(评测表)

评测表的详细信息见表 0-4。

表 0-4 hr_evaluation

表序号	4	表名		hr_evaluation		
含义	存储评测成绩					
序号	属性名称	含义	数据类型	长度	为空性	约束
1	evaluationid	评测 ID	int	11	not null	主键
2	evaluatorid	评测人 ID	int	11	not null	外键
3	evaluatedid	被评测人 ID	int	11	not null	外键
4	scord	评测成绩	int	11	not null	
5	evaluationdate	评测日期	varchar	14	not null	

5. hr_report(汇报表)

汇报表的详细信息见表 0-5。

表 0-5 hr_report

表序号	5	表名		hr_report		
含义	存储工作汇报内容					
序号	属性名称	含义	数据类型	长度	为空性	约束
1	reportid	汇报 ID	int	11	not null	主键
2	reporterid	汇报人 ID	int	11	not null	外键
3	content	汇报内容	varchar	500	not null	
4	reportdate	汇报日期	varchar	14	not null	

0.2.2 编写数据库脚本

下面给出创建企业 HR 管理系统(HRProject)和数据库表的 SQL 语句,读者在使用样例系统时,也可以直接运行配套资源中的建库脚本。

(1)hrproject 数据库。

```
Create database hrproject
```

(2)hr_user 表。

```
create table 'hr_user'(
  'userid' int(11) not null auto_increment,
  'empno' varchar(20) not null,
  'username' varchar(50) not null,
  'password' varchar(100) not null,
  'roleid' int(11) not null,
  'departid' int(11) not null,
  'salary' float(8, 2) null default null,
  primary key ('userid'),
  index 'roleid'('roleid'),
  index 'departid'('departid'),
```

```

        constraint 'hr_user_ibfk_1' foreign key ('roleid') references 'hr_role' ('roleid')
on delete cascade on update cascade,
        constraint 'hr_user_ibfk_2' foreign key ('departid') references 'hr_department'
('departid') on delete cascade on update cascade
    )

```

(3)hr_department 表。

```

create table 'hr_department'(
    'departid' int(11) not null,
    'name' varchar(255) not null,
    primary key ('departid')
)

```

(4)hr_role 表。

```

create table 'hr_role'(
    'roleid' int(11) not null,
    'rolename' varchar(255) not null,
    primary key ('roleid')
)

```

(5)hr_evaluation 表。

```

create table 'hr_evaluation'(
    'evaluationid' int(11) not null,
    'evaluatorid' int(11) not null,
    'evaluatedid' int(11) not null,
    'score' int(11) not null,
    'evaluationdate' varchar(14) not null,
    primary key ('evaluationid'),
    constraint 'hr_evaluation_ibfk_1' foreign key ('evaluatedid') references 'hr_user'
('userid') on delete cascade on update cascade,
    constraint 'hr_evaluation_ibfk_2' foreign key ('evaluatorid') references 'hr_user'
('userid') on delete cascade on update cascade
)

```

(6)hr_report 表。

```

create table 'hr_report'(
    'reportid' int(11) not null,
    'reporterid' int(11) not null,
    'content' varchar(500) not null,
    'reportdate' varchar(14) not null,
    primary key ('reportid'),
    index 'reporterid'('reporterid'),
    constraint 'hr_report_ibfk_1' foreign key ('reporterid') references 'hr_user'
('userid') on delete cascade on update cascade
)

```

0.2.3 导入数据库

(1)启动 Navicat 数据库管理工具,右击 localhost,在弹出的快捷菜单中选择“运行 SQL 文件”选项,如图 0-2 所示。

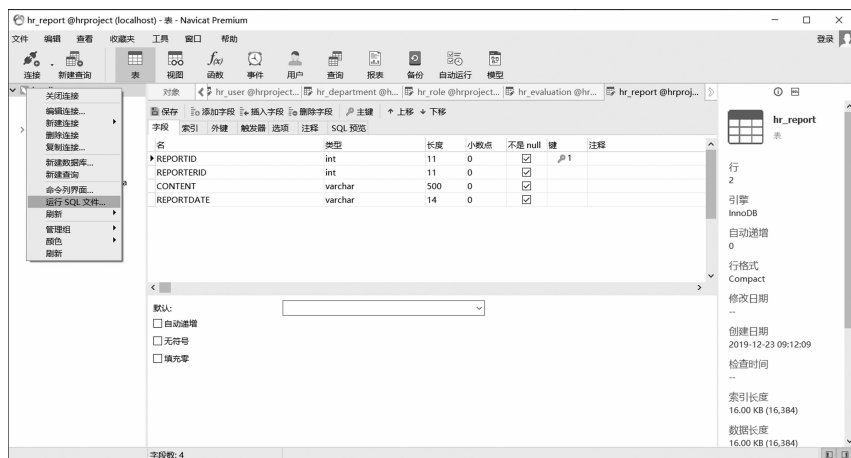


图 0-2 运行 SQL 文件

(2)选择企业 HR 管理系统对应的数据库文件 hrproject.sql,并单击“打开”按钮,如图 0-3 所示。

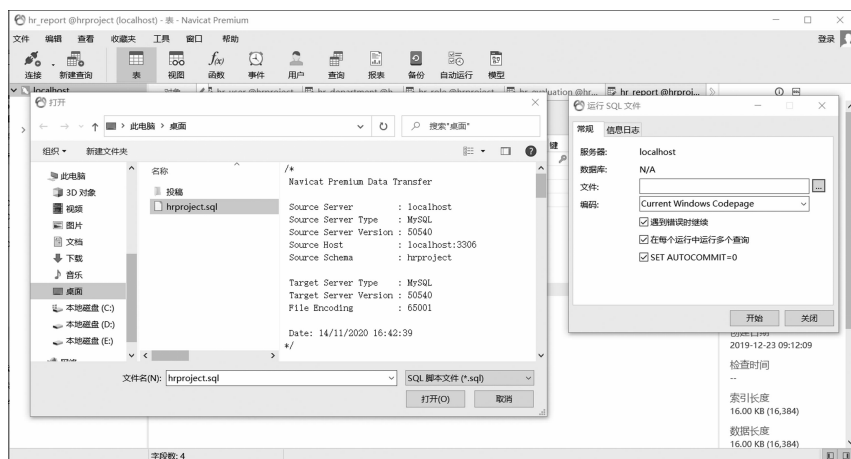


图 0-3 添加 SQL 脚本

(3)如图 0-4 所示,数据库已经添加至 MySQL 管理工具中,并初始化了部分测试数据。

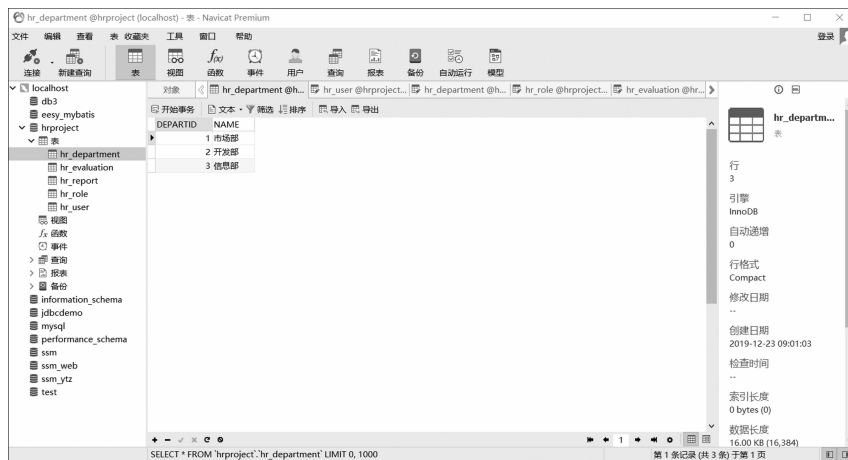


图 0-4 SQL 文件导入数据库

0.3 任务 2: 体验企业 HR 管理系统的功能

【任务目标】

通过使用本书所附的企业 HR 管理系统,体验企业 HR 管理系统的主要功能。

【任务要点】

配置运行本书所带的企业 HR 管理系统,注册后登录系统,完成一个完整的用户管理过程,体验企业 HR 管理系统的各个环节。

【任务实施】

- (1)进入用户登录界面。
- (2)注册成员工,默认注册的员工为普通员工。
- (3)登录企业 HR 管理系统。
- (4)各个角色的用户登录显示不同的界面。
- (5)普通员工的汇报操作,经理的查看汇报操作。
- (6)经理评测员工成绩,员工可以查看评测的成绩。
- (7)管理员用户修改员工角色。

0.3.1 登录界面

如前所述,企业 HR 管理系统运行使用的界面管理包含登录、注册、退出三个按钮,如图 0-5 所示。在“用户名”和“密码”文本框输入正确的内容后单击“登录”按钮,可以进入系统。

0.3.2 用户注册界面

单击“注册”按钮,用户可以注册成为企业的员工,用户注册时需要填写必要的资料和可选资料。只有企业员



图 0-5 用户登录界面

工才可以在企业 HR 管理系统中进行操作。用户注册界面如图 0-6 所示。



The image shows a 'User Registration' window with the following fields and buttons:

- Employee ID (员工编号): text input
- Employee Name (员工姓名): text input
- Password (密码): text input
- Confirm Password (确定密码): text input
- Department (部门): dropdown menu with '市场部' (Marketing Dept) selected
- Salary (薪资): text input
- Buttons: '注册' (Register) and '返回' (Return)

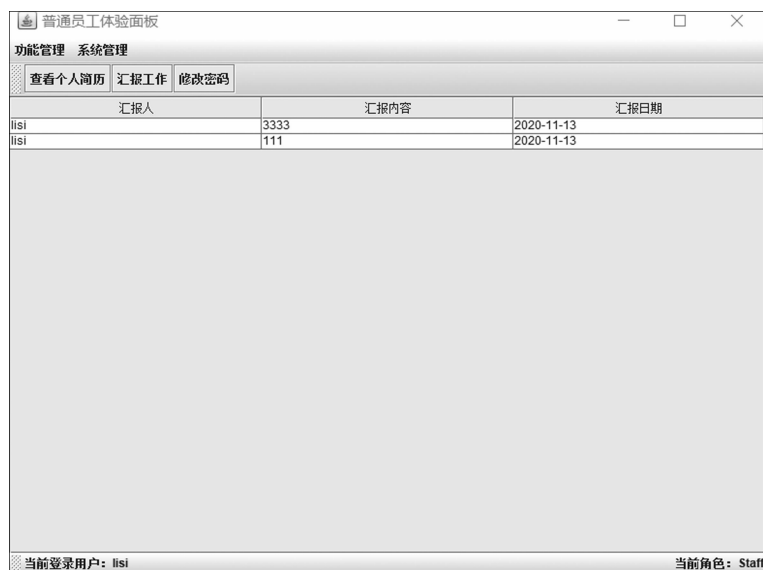
图 0-6 用户注册界面

0.3.3 用户角色界面

因为不同的用户角色所具有的管理权限不同,所以不同的用户登录后所显示的管理界面是不相同的。

1. 普通员工管理界面

普通员工管理界面如图 0-7 所示。



The image shows the 'Common Employee Management Interface' for a user named 'lisi'. It includes a navigation bar with '功能管理' (Function Management) and '系统管理' (System Management). Below the navigation bar are three tabs: '查看个人简历' (View Personal Resume), '汇报工作' (Report Work), and '修改密码' (Change Password). The '汇报工作' tab is active, displaying a table with the following data:

汇报人	汇报内容	汇报日期
lisi	3333	2020-11-13
lisi	111	2020-11-13

At the bottom of the interface, it shows '当前登录用户: lisi' (Current logged-in user: lisi) and '当前角色: Staff' (Current role: Staff).

图 0-7 普通员工管理界面

2. 经理管理界面

经理管理界面如图 0-8 所示。

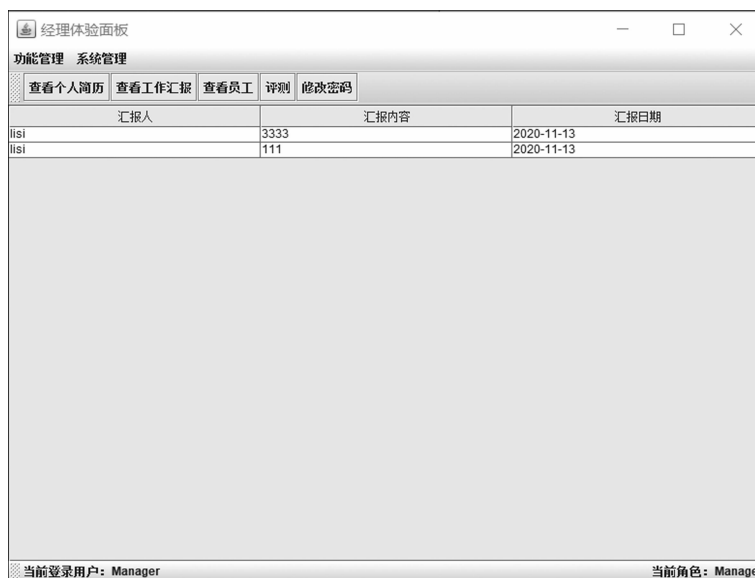


图 0-8 经理管理界面

3. 管理员管理界面

管理员管理界面如图 0-9 所示。



图 0-9 管理员管理界面

0.3.4 汇报工作界面

(1) 普通员工在进行工作汇报时可以单击“汇报工作”按钮,进行当日的工作总结,如图 0-10 所示。

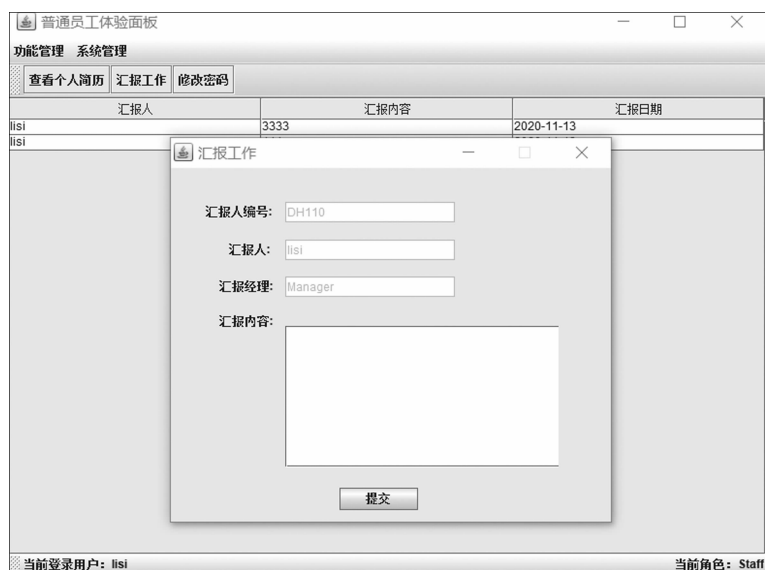


图 0-10 普通员工汇报工作界面

(2)普通员工汇报完当日工作,切换经理账户进行登录可以查看其部门员工的汇报信息,如图 0-11 所示。

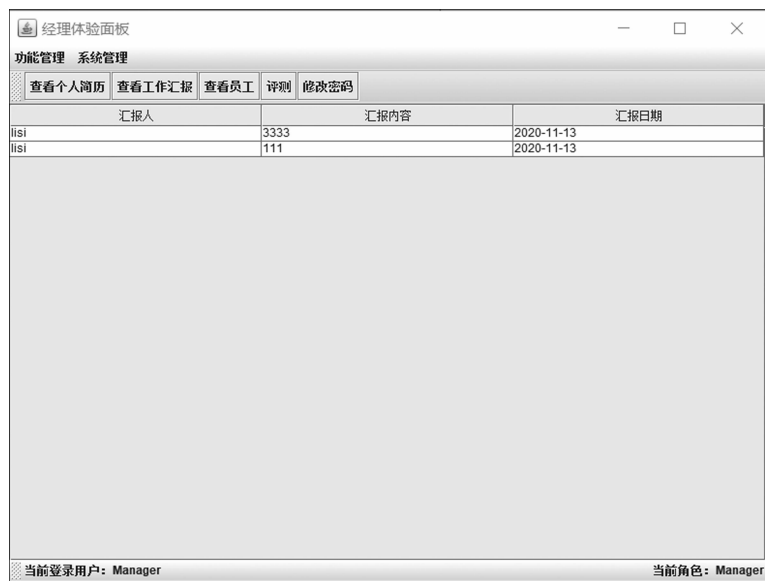


图 0-11 经理查看工作汇报界面

0.3.5 评测员工成绩界面

(1)经理定期对其部门员工进行成绩的评定,以便激励员工更好的工作,具体界面展示如图 0-12 所示。

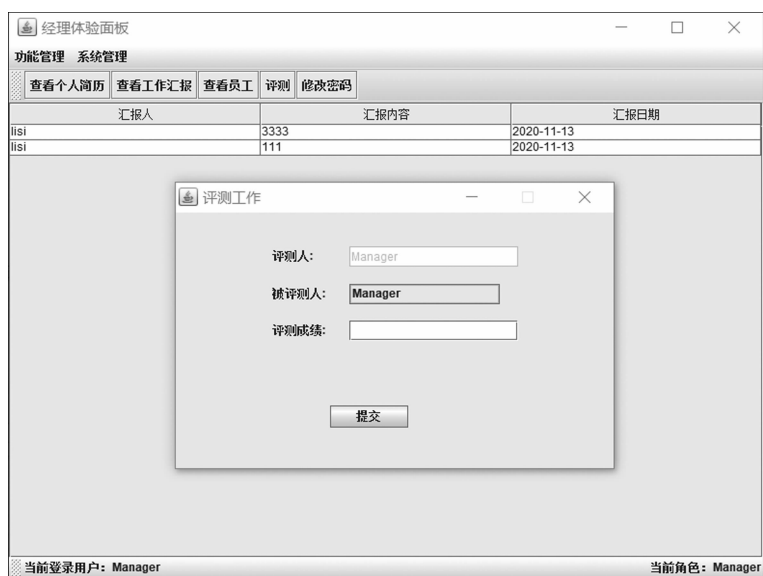


图 0-12 经理评测员工成绩界面

(2) 经理评测成绩后, 普通员工可以登录系统进行查看, 如图 0-13 所示。

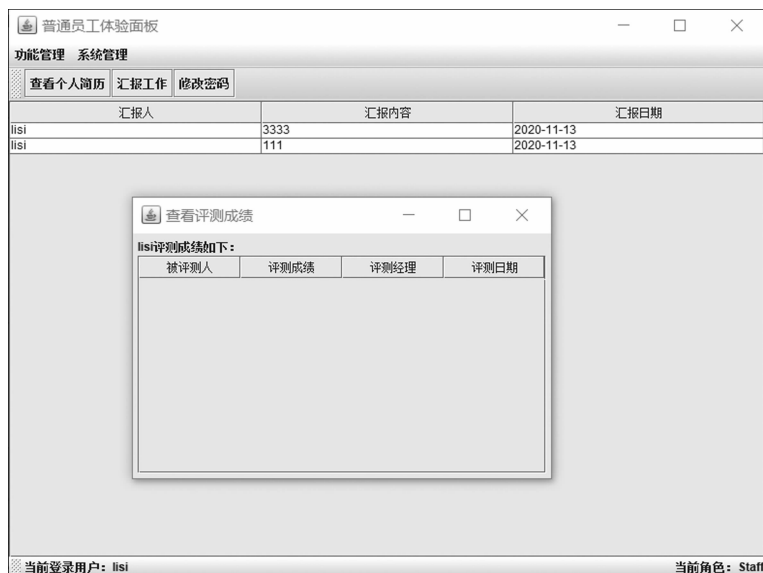


图 0-13 普通员工查看评测成绩界面

0.3.6 更改角色

管理员用户可以修改用户角色, 如图 0-14 所示。



图 0-14 管理员用户修改用户角色界面

0.3.7 修改密码

普通员工、经理、管理员三种角色用户都可以对自己的登录密码进行修改,如图 0-15 所示。

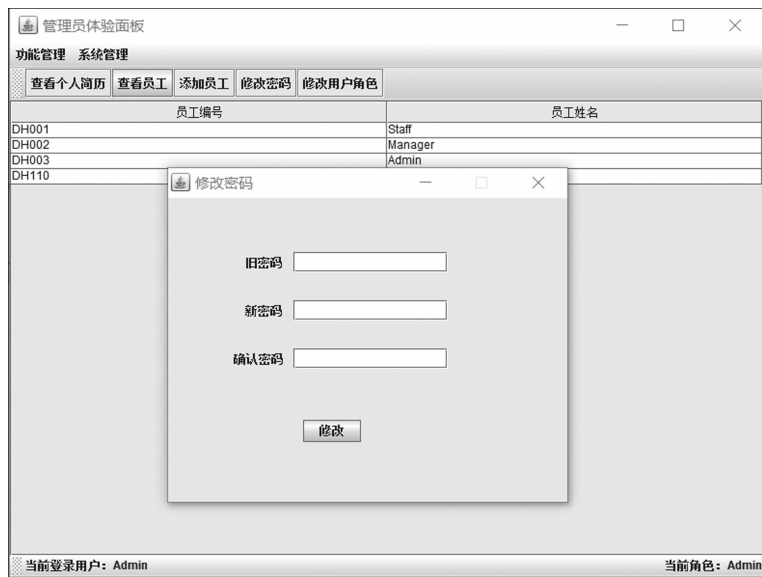


图 0-15 用户修改密码界面

0.3.8 查看个人简历

普通员工、经理、管理员三种角色用户登录后都可以查看个人信息,如图 0-16 所示。



图 0-16 用户查看个人信息界面

0.3.9 系统使用说明

1. 系统

本案例是在 Windows 10 家庭版操作系统下开发的,程序测试环境为 Windows 10 家庭版操作系统。用户在 Windows 10 家庭版操作系统下正确配置程序运行所需的环境后,完全可以运行本案例。系统具体配置如下。

(1)硬件平台。

①CPU:英特尔 i5-2430M 2.4 GHz。

②内存:2 GB 以上。

(2)软件平台。

①操作系统:Windows 10。

②数据库:MySQL。

③开发工具包:JDK 1.8。

④开发环境:Eclipse。

2. 源程序使用方法

如果用户要使用企业 HR 管理系统源程序,除了需要满足上面要求的计算机配置外,还需要导入源程序文件。

(1)打开 Eclipse,执行“File”→“Import”命令,导入项目工程文件,如图 0-17 所示。

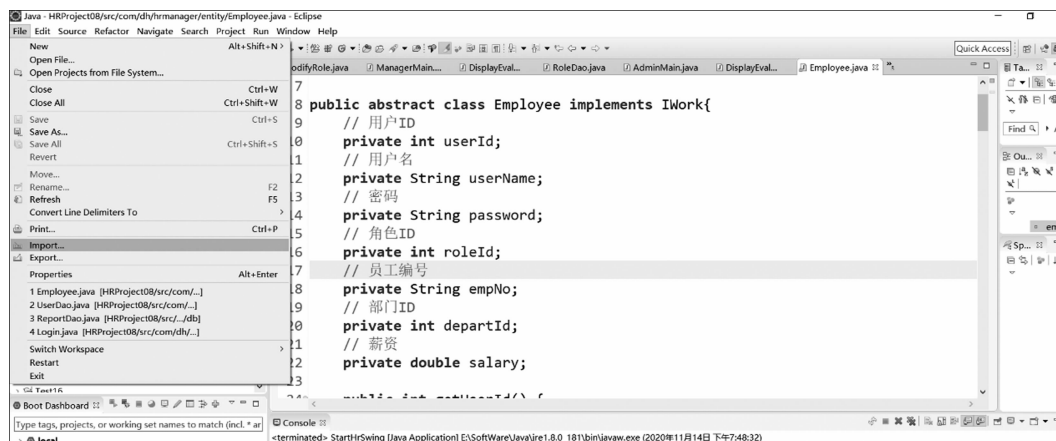


图 0-17 导入项目工程文件

(2)如图 0-18 所示,在打开的“Select”界面中选择“Existing Projects into Workspace”选项,单击“Next”按钮,在打开的“Import Projects”界面中选择项目所在文件夹,如图 0-19 所示。

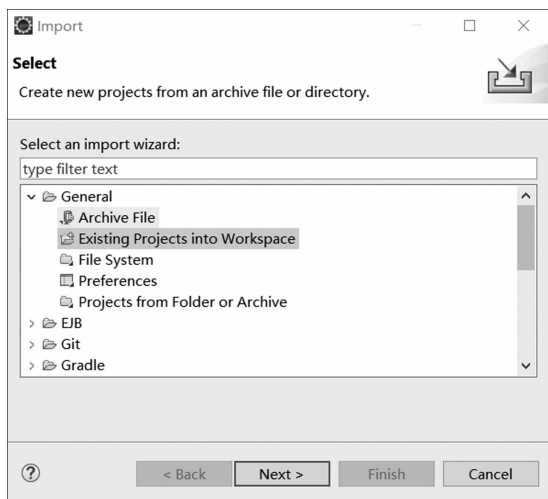


图 0-18 选择导入方式

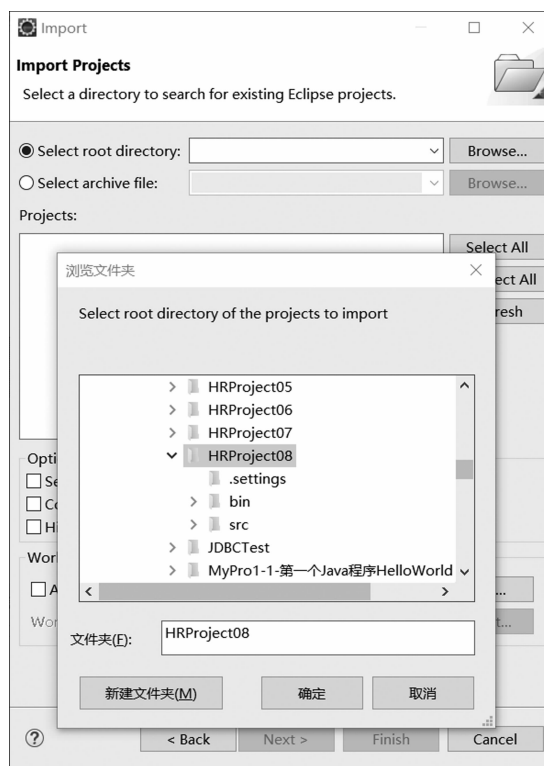


图 0-19 选择导入的项目工程文件夹

最后将项目添加至 Eclipse 项目工程面板中,如图 0-20 所示。

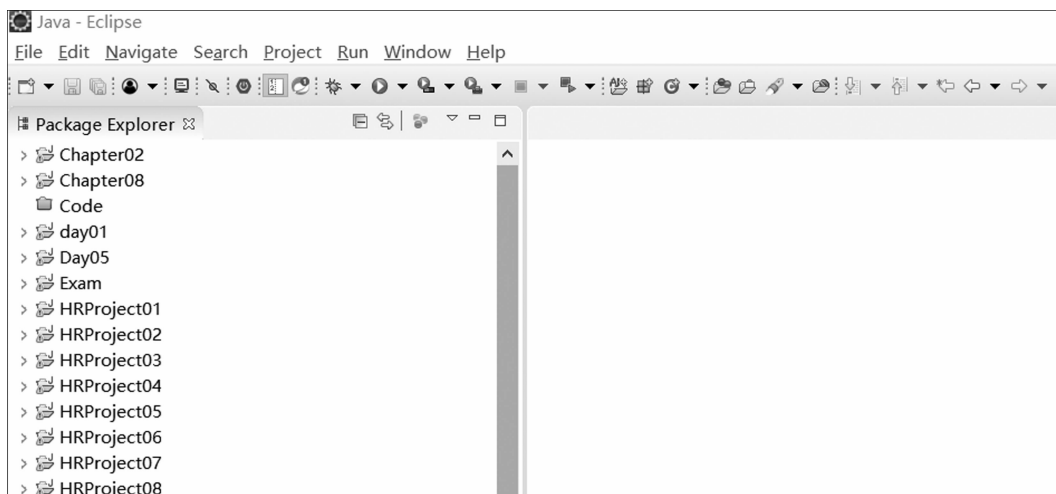


图 0-20 导入成功界面

(3)单击“运行”按钮  启动项目，系统启动界面见图 0-5。

本章小结

本章对本书所涉及的企业 HR 管理系统进行了简要概述，对系统各功能模块和系统数据库设计进行了说明。通过创建数据库、导入项目等操作步骤，启动管理系统，体验企业 HR 管理系统的功能，让读者真正体会软件开发的整体流程，以便将后续知识进行贯穿、融合。

Java 开发入门

 学习目标

了解 Java 语言的来历,掌握 Java 语言的特点;认识 Java 虚拟机,即任何一种可以运行 Java 程序的软件都可以称为 Java 虚拟机(Java virtual machine,JVM);掌握面向对象程序设计的特点;掌握抽象、封装、类、对象的概念等。

1.1 Java 的来历

Java 语言与咖啡有着不解之缘。

2000 年,JavaOne 国际会议大厅热闹非凡,一阵阵浓郁的咖啡香气扑鼻。从世界各地汇集到旧金山参加会议的 Java 精英们兴奋异常,排着长队,等待得到一杯由 Java 语言控制的咖啡机煮制的免费咖啡。这是一款经过特殊设计的全透明咖啡机,当按下按钮时,咖啡豆从玻璃管输送到研磨器,磨制后的咖啡粉由另一个玻璃管送到煮制机,沸腾的水滴入咖啡粉中,一杯醇香四溢的咖啡就完成了。

关于 Java 的命名由来,在美国硅谷地区有许多不同版本的故事。Sun Microsystems 公司的创建者和前首席执行官(CEO) James Gosling 也加入了讲述这个故事行列。最初这个为电视机顶盒所设计的语言在公司内部一直称为 Green 项目。这个新语言需要一个名字。Sun Microsystems 公司的创建者 James Gosling 一直在考虑这个问题。有一天,他注意到自己办公室外一棵茂密的橡树 Oak,这是一种在硅谷很常见的树。所以他将这个新语言命名为 Oak。但 Oak 是一家注册公司的名字,这个名字不可能再用了。James Gosling 通过市场部门请来了一个命名顾问,召开命名征集会。当时,因为没有合适的名字,他们不得不延后这个语言的面世。在命名征集会上,大家提出了很多名字。最后按大家的评选次序将十几个名字排列成表,上报给商标律师。排在第一位的是 Silk(丝绸)。尽管大家都喜欢这个名字,但遭到 James Gosling 的坚决反对而作罢。排在第二和第三的都没有通过律师这一关。James Gosling 最喜欢的就是排在第三位的 Lyric(抒情诗)。只有排在第四位的名字得到了所有人的认可和律师的通过。这个名字就是 Java。第一个提议名字 Java 的是 Mark Opperman,Mark Opperman 是在一家咖啡店与同事品尝咖啡时得到灵感的。Java 是印度尼西亚爪哇岛的英文名称,因盛产咖啡而闻名。国外的许多咖啡店用 Java 来命名或宣传,以彰显其咖啡的品质。Java 语言中的许多库类名称多与咖啡



视频

Java 的来历和
Java 语言的
特点

有关,如 JavaBeans(咖啡豆)、NetBeans(网络豆)、ObjectBeans(对象豆)等。从此一个既好听又好记、具有强大生命力的编程语言 Java 诞生了。

Java 是一种可以编写跨平台应用程序的面向对象的程序设计语言,Java 技术具有卓越的通用性、高效性、平台移植性和安全性,广泛应用于个人计算机、数据中心、游戏控制台、科学超级计算机、移动电话和互联网,同时拥有全球最大的开发者专业社群。在全球云计算和移动互联网的产业环境下,Java 更具备显著优势和广阔前景。

1.2 Java 语言的特点

1. 易于学习

Java 语言很简单,但这里所说的简单,主要是针对熟悉类似 C++ 语言的程序设计人员来说的,因为它的语法与 C++ 非常相似,但是它摒弃了 C++ 中许多低级、困难、容易混淆、容易出错或不经常使用的功能,如运算符重载、指针运算、程序的预处理、结构、多重继承及其他一系列内容,这样便使其比其他许多编程语言更易于学习。利用 Java 语言还能够编写出非常复杂的系统,且运行时占用很少的内存资源。

2. 高效率的执行方式

用 Java 语言编写的源程序是采用先经过编译器编译,再利用解释器解释的方式来运行的。它综合了解释性语言与编译语言的众多优点,使其执行效率较以往的程序设计语言有了大幅度的提高。

3. 平台无关性

Java 源程序经过编译器编译,会被转换成一种称为字节码(byte codes)的目标程序。字节码的最大特点是可以跨平台运行,即程序设计人员常说的“编写一次,到处运行”,正是这一特性使得 Java 得到迅速普及。

4. 分布式

分布式主要是指数据分布和操作分布两层意思,数据分布是指数据可以分散在网络中不同的主机上,操作分布是指把一个任务分散在不同的主机上进行处理。Java 从诞生起就与网络联系在一起,它强调网络特性,内置 TCP/IP、HTTP、FTP 协议类库,支持远程方法调用,便于开发网上应用系统。

5. 安全性

Java 平台采用了域管理方式的安全模型,无论是本地代码还是远程代码都可以通过配置的策略设定可访问的资源域。当 Java 字节码进入专门处理该内容的程序时,首先必须经过字节码校验器的检查,然后 Java 解释器将决定程序中类的内存布局,随后,类装载器负责把来自网络的类装载到单独的内存区域,避免应用程序之间相互干扰破坏。最后,客户端用户还可以限制从网络上装载的类只能访问某些文件系统。上述几种机制结合起来,使得 Java 成为安全的编程语言。

6. 可靠性

Java 程序设计语言要求显式的方法声明,保证了编译器可以发现方法调用错误;Java 不支持指针,杜绝了对内存的非法访问;Java 的垃圾自动收集机制,防止了程序员可能忘记释放原来

分配的内存或者释放了其他正在使用的内存而引起的系统严重出错,这些措施保证了 Java 程序的可靠性。

7. 多线程

Java 的多线程机制使应用程序中的线程能够并发执行,且其同步机制保证了对共享数据的正确操作。通过使用多线程,程序设计者可以分别用不同的线程完成特定的行为,而不需要采用全局的事件循环机制,这样就很容易在网络上实现实时交互行为。

8. 丰富的应用程序接口类库

Java 开发工具包中的类库很丰富,程序员的开发工作可以在一个更高的层次上展开,这也是 Java 受欢迎的重要原因之一。Java 同时为用户提供了详尽的应用程序接口(application program interface,API)文档说明。

1.3 Java 虚拟机

Java 语言源程序是先经过编译器编译,再利用解释器解释的方式来运行的。Java 程序的开发及运行周期如图 1-1 所示。



图 1-1 Java 程序的开发及运行周期

对于 Java 运行平台无关的特点,我们可以通俗地理解为:在常规计算机运行环境中,一定存在多种类型的 Java 解释程序帮助运行 Java 程序。任何一种可以运行 Java 程序的软件都可以称为 Java 虚拟机,即 Java 虚拟机就是一个将字节码指令映射为对应物理操作系统指令的程序。

1.4 面向对象程序设计

面向对象程序设计(object oriented programming)是一种程序设计范型,同时也是一种程序开发方法。对象指的是类的实例。它将对象作为程序的基本单元,将程序和数据封装其中,以提高软件的重用性、灵活性和扩展性。

面向对象程序设计可以看作一种在程序中包含各种独立而又互相调用的对象的思想,这与传统的思想刚好相反:传统的程序设计主张将程序看作一系列函数的集合,或者直接就是一系列对计算机下达的指令。面向对象程序设计中的每一个对象都应该能够接受数据、处理数据并将数据传达给其他对象,因此它们都可以被看作一个小型的机器,即对象。

1.4.1 抽象

抽象概念是传统逻辑用以支撑现实中没有一个或一类具体事物与之相适应(其外延不是一个或一类事物),而仅以依存于某个或某类事物的性质和事物与事物之间的关系为其反映对象的概念。与具体概念相对。

抽象,顾名思义,就是抽掉了具体形象的东西。例如,人就可以说是一个相对抽象的概念,因为一个具体的人有性别、年龄、身高、相貌、社会关系等具体特征,但抽象的人就是不包含这些具体特征的一个概念,因此说它是抽象的;而具体的人的概念应该是张三、李四等。因此,抽象概念一般是指某一类事物,而不是某一个事物。

1.4.2 封装

在面向对象设计方法中,封装是指将数据及对数据的操作方法包装或隐藏起来的方法。封装只需知道如何使用,而不需要了解其内部实现细节。例如,对于电视机,使用者只需操作开关键、频道键和声音键等即可,既无需了解电视机的内部工作原理,也不需要去操控电视机内部的元件。这不仅给人们带来了使用上的方便,也避免了安全事故发生(如触电)。适当的封装可以让程序代码更容易理解与维护,也加强了程序代码的安全性。要访问该封装体内的代码和数据,必须通过严格的接口控制。电视机外的那些操作键,可以看作程序片段的接口。通俗地说,封装就是需要让用户知道的暴露出来,不需要让用户了解的全部隐藏起来。

1.4.3 类与对象的概念

1. 对象

对象是描述客观事物的一个实体,几乎所有东西都可以看作对象。对象可以是有形的,如坦克、学生、文件,也可以是无形的,如一个方案、头脑中的一个想法。人类认识世界,其实就是面向对象的。程序中的对象来源于生活,现实生活中的某个实体经过抽象、建模后就变成程序中的一个对象,即用特定符号描述的对象。对象都有自己的特性,称为属性(静态特征)和方法(动态特征)。例如,学生的属性有学号、姓名、性别等,其方法有学习、选课等。由于对象和现实世界的实体有联系紧密,所以面向对象编程更加贴近人们的思维习惯。

2. 类

物以类聚,人以群分。现实生活中的一组相似对象(具有共同的特征)可以归为一类。例如,哺乳动物是一个类,它们的共性为:有脊椎、胎生、用肺呼吸、恒温等,一只具体的猫则是一个对象。生物学中的分类法(如林奈氏分类法)就是典型的对生物对象进行归纳成类的方法。在程序设计中,把一组相似对象的共同特征抽象出来并存储在一起,就形成了类。每一只哺乳动物都具有哺乳动物类所规定的属性和方法,但是其属性值可能不同,如人的恒温值和猫的恒温值就不同。

从具体对象到类实际上是一个概括的过程,把对象的共同特征抽取出来形成类,这个过程称为抽象。通俗地解释,抽象即“抽出像的部分”。抽象包括数据抽象和方法抽象,形成属性和方法两方面的共同特征。

类也可以看作一个模板或蓝图,系统根据类的定义来造出对象。我们如何制造一辆汽车?类就是汽车的蓝图(图纸),规定了汽车的详细信息,根据图纸可以将汽车制造出来。一个模板可以生产很多产品,一张图纸也可以制造出很多辆汽车。

对象是具体的事物,称为 object 或 instance(实例)。

类是对对象的抽象,称为 class。

对象和类的关系是特殊到一般,具体到抽象。

类可以看成一类对象的模板,对象可以看成该类的一个具体实例。



视频
类与对象

类是用于描述同一类型对象的一个抽象的概念,类中定义了这一类对象所应具有静态属性和动态属性。

对象是 Java 程序的核心,在 Java 程序中“万事万物皆对象”。

J2SDK 提供了很多类供编程人员使用,编程人员也可定义自己的类。

1.5 任务:创建企业 HR 管理系统中的类和对象

【任务目标】

Java 环境下,企业 HR 管理系统中的类和对象的具体实现。

【任务要点】

本书以企业 HR 管理系统为例,根据类和对象的基本概念,学习企业 HR 管理系统中类和对象的创建。

【任务实施】

企业 HR 管理系统中主要包括的类有员工类 Employee、汇报类 Report、评价类 Evaluation。其中,员工类 Employee 主要存储员工信息及其操作,员工类的属性有员工 ID、员工编号、姓名、密码、角色 ID、工资、部门 ID 等。员工类是经理类、普通员工类、管理员类的基类,这些继承员工基类的相关属性,系统中具体的实现代码如下:

1) 员工类

```
public abstract class Employee implements IWork{
    // 用户 ID
    private int userId;
    // 用户名
    private String userName;
    // 密码
    private String password;
    // 角色 ID
    private int roleId;
    // 员工编号
    private String empNo;
    // 部门 ID
    private int departId;
    // 薪资
    private double salary;
    public int getUserId() {
        return userId;
    }
    public void setUserId(int userId) {
        this.userId = userId;
    }
    public String getUserName() {
```

```
        return userName;
    }

    public void setUserName(String userName) {
        this.userName = userName;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public int getRoleId() {
        return roleId;
    }

    public void setRoleId(int roleId) {
        this.roleId = roleId;
    }

    public String getEmpNo() {
        return empNo;
    }

    public void setEmpNo(String empNo) {
        this.empNo = empNo;
    }

    public int getDepartId() {
        return departId;
    }

    public void setDepartId(int departId) {
        this.departId = departId;
    }

    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    public Employee() {
    }

    public Employee(int userId, String empNo, String userName, String password, int
departId, int roleId, double salary) {
        this.userId = userId;
        this.empNo = empNo;
```

```

        this.userName = userName;
        this.password = password;
        this.departId = departId;
        this.roleId = roleId;
        this.salary = salary;
    }
    /* *
    * 显示员工信息
    */
    public void displayInfo() {
        String position = null;
        if (roleId == 1)
            position = "Staff";
        if (roleId == 2)
            position = "Manager";
        if (roleId == 3)
            position = "Admin";
        String department = null;
        if (departId == 1)
            department = "市场部";
        if (departId == 2)
            department = "开发部";
        if (departId == 3)
            department = "信息部";
        System.out.println("\t\t 当前用户信息\n\n");
        System.out.println(" * * * * * ");
        System.out.println("\t\t 雇员编号:" + empNo);
        System.out.println("\t\t 姓名:" + userName);
        System.out.println("\t\t 职务:" + position);
        System.out.println("\t\t 部门:" + department);
        System.out.println("\t\t 薪水:" + String.valueOf(salary));
        System.out.println(" * * * * * ");
    }
    /* *
    * 修改密码
    *
    * @param newPassword
    */
    public void modifyPassword(String newPassword) {
        this.password = newPassword;
    }

```

```

        System.out.println("密码修改成功");
    }
    /* *
     * 查看工资范围,抽象方法
     */
    public abstract void displaySalaryRange();
}

```

2) 普通员工类

```

public class Staff extends Employee {
    public Staff() {
        super();
    }

    public Staff(int userId, String empNo, String userName, String password, int
departId, int roleId, double salary) {
        super(userId, empNo, userName, password, departId, roleId, salary);
    }

    HrHelper helper = new HrHelper();
    /* *
     * 添加汇报
     */
    public void addReport() {
        System.out.println("请输入汇报信息\n");
        Scanner input = new Scanner(System.in);
        String content = input.next();
        // 实例化 report 对象
        Report report = new Report();
        // 汇报 ID,每次在上一次基础上加 1
        report.setReportId(helper.getReportsCount() + 1);
        report.setContent(content);
        // 设置汇报人 ID
        report.setReporterId(Data.currentEmployee.getUserId());
        // 加入汇报存储数组
        helper.addReport(report);
    }

    @Override
    public void displaySalaryRange() {
        // TODO Auto-generated method stub
        System.out.println("Staff 薪资范围为:3000~5000");
    }
}

```

```

@Override
public void doWork() {
    // TODO Auto-generated method stub
    System.out.println(helper.getDate() + ",Staff 工作任务是:");
    System.out.println("=====");
    System.out.println("\t 到学校进行讲座!");
    System.out.println("=====");
}
// 显示所有评估
public void displayEvaluation() {
    helper.displayEvaluations();
}
// 显示所有评估,由高到低排序
public void displayEvaluationDesc() {
    helper.displayEvaluationsDesc();
}
}

```

3) 经理类

```

public class Manager extends Employee {
    // 实例化业务逻辑类
    HrHelper helper = new HrHelper();
    /* *
    * 显示所有汇报
    */
    public void displayReports() {
        helper.displayReports();
    }
    public Manager() {
        super();
    }
    public Manager(int userId, String empNo, String userName, String password, int
departId, int roleId, double salary) {
        super(userId, empNo, userName, password, departId, roleId, salary);
    }
    @Override
    public void displaySalaryRange() {
        // TODO Auto-generated method stub
        System.out.println("Manager 薪资范围为:5000~8000");
    }
}

```

```

@Override
public void doWork() {
    // TODO Auto-generated method stub
    System.out.println(helper.getDate() + ",Manager 工作任务是:");
    System.out.println("=====");
    System.out.println("\t制订出差计划!");
    System.out.println("=====");
}
// 添加评测
public void addEvaluation(Evaluation evaluation) {
    helper.addEvaluation(evaluation);
}
}

```

4) 管理员类

```

public class Admin extends Employee {
    HrHelper helper = new HrHelper();
    /* *
     * 修改员工角色
     * @param empNo, 员工编号
     * @param role, 角色名称, 只能是 Staff、Manager、Admin
     */
    public void modifyEmployeeRole(String empNo, String role) {
        helper.modifyEmployeeRole(empNo, role);
    }
    public Admin() {
        super();
    }
    public Admin(int userId, String empNo, String userName, String password, int
departId, int roleId, double salary) {
        super(userId, empNo, userName, password, departId, roleId, salary);
    }
    @Override
    public void displaySalaryRange() {
        // TODO Auto-generated method stub
        System.out.println("Admin 薪资范围为:4000~6000");
    }
    /* *
     * 显示员工信息
     * @param employee

```

```
    */  
    public void displayEmployeeInfo(Employee employee) {  
        helper.displayEmployeeInfo(employee);  
    }  
    @Override  
    public void doWork() {  
        // TODO Auto-generated method stub  
        System.out.println(helper.getDate()+"Admin 工作任务是:");  
        System.out.println("=====");  
        System.out.println("\t维护员工基本信息!");  
        System.out.println("=====");  
    }  
}
```

本章小结

本章深入学习了 Java 语言的来历、Java 语言的特点、Java 虚拟机的概念和作用及面向对象程序设计的相关知识,主要包括易于学习、高效率的执行方式、平台无关性、分布式、安全性、可靠性、多线程、丰富的 API 类库等,也介绍了面向对象的程序设计、类、对象的概念等知识。

课后练习

1. Java 语言有哪些特点与优势?
2. 什么是 Java 虚拟机? 它的主要作用是什么?
3. 什么是类? 什么是对象? 举例说明。