



于立萍 赵小芳◎编著

Java

网络程序设计实验指导教程

Java 网络程序设计 实验指导教程

于立萍 赵小芳 编著

简 介

《java 网络程序设计实验指导教程》是 java 网络程序设计课程的实验指导书,用于帮助读者学习 Java 程序设计,并侧重网络编程的训练,培养基本的编程能力和项目的实践能力。

本书共安排 10 个实验,分别介绍了 Java 开发环境的搭建,Java 程序基础,面向对象程序设计基础,面向对象程序设计进阶,Java GUI 编程,多线程程序设计,基于套接字的网络通信,JDBC 编程以及涵盖 JSP、Servlet 的 Java Web 应用等内容。

本书的实验是专门针对 Java 网络程序设计课程的补充练习,也可作为各类高等院校、高职高专计算机类专业和各种培训班的辅助实验教材,还可供广大科技人员和 java 技术爱好者参考。

前 言

随着 Internet 在我国的迅速普及和广泛应用, 社会对网络编程人才的需求日益增加, 培养网络编程人员刻不容缓。Java 是目前国内外广泛应用的新一代面向对象的程序设计语言, 特别适合于 Internet 应用程序的开发。为此, 我们编写了《Java 网络程序设计实验指导教程》, 该书注重实用性, 旨在培养学生基本的编程能力和项目实践能力。

本书是为 Java 网络程序设计课程配置的教学辅导教材。全书主要包括 Java 开发环境的搭建, Java 程序基础, 面向对象程序设计基础, 面向对象程序设计进阶, Java GUI 编程, 多线程程序设计, 基于套接字的网络通信, JDBC 编程以及涵盖 JSP、Servlet 的 Java Web 应用等 10 个实验。具体如下:

实验 1: 搭建 Java 开发环境。本实验为开发和编写 Java 网络应用程序做一些准备工作。主要包括 JDK 的下载、安装与配置, Eclipse 的下载与安装等内容。

实验 2: 结构化编程。实验内容主要包括 Java 的基本数据类型、结构化程序设计等。

实验 3: 面向对象编程。实验内容主要包括类的封装、继承和多态等。

实验 4: 面向对象编程进阶。实验设计了一个简单的记账系统, 通过本实验进一步强化面向对象程序设计核心概念的掌握, 同时培养学生项目实践能力。

实验 5: GUI 编程。实验内容主要包括基本控件的使用、窗体的布局和事件的响应等内容。

实验 6: 多线程编程。介绍了如何实现基于多线程的生产者与消费者关系。

实验 7: 网络通信。实验设计了一个基于套接字的 C/S 模式的聊天程序。通过该项

目整合 GUI 编程、多线程编程和网络编程的基本内容,进一步培养学生工程实践的能力。

实验 8:JDBC 编程。主要介绍数据库的连接,通过 JDBC 实现数据库基本的增、删、改、查操作以及通过构建可滚动可更新的结果集,完成数据库的浏览和更新操作。

实验 9:Java Web 应用—JSP。实验设计了一个基于 JSP 的雇员信息管理项目。该项目主要包括管理员的登录、雇员信息的显示等功能。

实验 10:Java Web 应用—Servlet。实验设计了一个基于 MVC 模式的雇员信息管理项目。该项目采用 Servlet 作为控制器,实现了管理员的登录、雇员信息的分页显示和删除等功能。

本书由山东工商学院计算机科学与技术学院于立萍、赵小芳老师编著。本书在写作过程中,也得到了山东工商学院相关领导的指导和支持。为此对一切曾经支持和帮助过我们的领导和朋友表示真挚的感谢。

由于时间仓促和水平有限,书中难免有不妥之处,敬请读者不吝赐教。

编著者

2013 年 4 月

目 录 contents

前言

实验 1 搭建 Java 开发环境

- 1.1 实验目的与要求
- 1.2 实验内容及步骤
 - 1.2.1 JDK 软件包的下载与安装
 - 1.2.2 设置 Java 运行环境
 - 1.2.3 Eclipse 软件包的下载与安装
 - 1.2.4 编写并运行一个显示“Hello World”的 Java 应用程序

实验 2 结构化编程

- 2.1 实验目的与要求
- 2.2 预习内容
- 2.3 实验内容
 - 2.3.1 基本数据类型
 - 2.3.2 循环结构
- 2.4 实验步骤
 - 2.4.1 基本数据类型
 - 2.4.2 循环结构
- 2.5 思考与讨论

实验 3 面向对象编程

- 3.1 实验目的与要求
- 3.2 预习内容
- 3.3 实验内容
 - 3.3.1 类的定义
 - 3.3.2 类的多态
- 3.4 实验步骤
 - 3.4.1 类的定义
 - 3.4.2 类的多态
- 3.5 思考与讨论

实验 4 面向对象编程进阶

- 4.1 实验目的与要求
- 4.2 预习内容
- 4.3 实验内容
 - 4.3.1 欢迎页面
 - 4.3.2 登录页面
 - 4.3.3 主页面
 - 4.3.4 个人信息页面
 - 4.3.5 添加账目
 - 4.3.6 查询账目页面
- 4.4 实验步骤
 - 4.4.1 构建相关的实体类
 - 4.4.2 设计公共功能类
 - 4.4.3 主程序
 - 4.4.4 登录页面
 - 4.4.5 主页面

- 4.4.6 个人信息页面
- 4.4.7 添加账目页面
- 4.4.8 账目查询页面
- 4.5 思考与讨论

实验 5 GUI 编程

- 5.1 实验目的与要求
- 5.2 实验内容
- 5.3 实验步骤
- 5.4 思考与讨论

实验 6 多线程编程

- 6.1 实验目的与要求
- 6.2 预习内容
- 6.3 实验内容
- 6.4 实验步骤
 - 6.4.1 定义产品类
 - 6.4.2 定义仓库类
 - 6.4.3 生产者线程
 - 6.4.4 消费者线程
 - 6.4.5 测试 ProducerConsumer 类
 - 6.4.6 运行结果
- 6.5 思考与讨论

实验 7 网络通信

- 7.1 实验目的与要求

7.2 预习内容

7.3 实验内容

7.4 实验步骤

7.4.1 服务器端

7.4.2 客户端

7.5 思考与讨论

实验 8 JDBC 编程

8.1 实验目的与要求

8.2 预习内容

8.3 实验内容

8.4 实验步骤

8.4.1 Eclipse 中通过 Data Source Explorer 连接、操作数据源

8.4.2 建立数据表 employees、managers 和 departments

8.4.3 JDBC 基本操作及测试

8.4.4 ResultSet 的基本操作

8.5 思考与讨论

实验 9 Java Web 应用—JSP

9.1 实验目的与要求

9.2 预习内容

9.3 实验内容

9.3.1 Tomcat 服务器的安装与配置

9.3.2 基于 JSP 实现管理员的登录及雇员信息的显示

9.4 实验步骤

9.4.1 Tomcat 服务器的安装与配置

9.4.2 基于 JSP 实现管理员的登录及雇员信息的显示

9.5 思考与讨论

实验 10 Java Web 应用—Servlet

10.1 实验目的与要求

10.2 预习内容

10.3 实验内容

10.3.1 管理员登录

10.3.2 雇员信息的维护

10.4 实验步骤

10.4.1 实验准备

10.4.2 项目的文件结构

10.4.3 Servlet 的配置

10.4.4 用户登录功能的实现

10.4.5 分页功能的实现

10.4.6 雇员信息显示功能的实现

10.4.7 删除雇员功能的实现

10.4.8 基于 Servlet Filter 解决中文乱码问题

10.5 思考与讨论

实验 1 搭建 Java 开发环境

1.1 实验目的与要求

- (1)掌握 Java 的 JDK 软件包的下载与安装;
- (2)掌握设置 Java 运行环境;
- (3)掌握 Eclipse 软件包的下载与安装;
- (4)编写并运行一个显示“Hello World”的 Java 应用程序。

1.2 实验内容及步骤

1.2.1 JDK 软件包的下载与安装

Java 的 JDK 开发包是免费软件,可以直接到 Oracle 的 Java 技术支持网站(<http://www.oracle.com/technetwork/java/index.html>)进行下载。JDK 包含了完整的 Java 程序开发工具。

Java 的安装很简单,双击即可。安装过程可以选择安装目录及 Java 运行目录(即 JRE)。

1.2.2 设置 Java 运行环境

环境变量用于设置 Java 编译与解释等工具的路径,选择 Java 安装目录下的 bin 目录。Path 设置如图 1-1 所示。



图 1-1 编辑环境变量 path

Classpath 用于设置应用程序类路径和扩展的应用程序类路径，指定当前运行目录和 Java 安装目录下的 lib 目录。Classpath 设置如图 1-2 所示。



图 1-2 编辑环境变量 classpath

1.2.3 Eclipse 软件包的下载与安装

Eclipse 是一款开源的、基于 Java 的可扩展的集成开发平台。可以直接到 Eclipse 的官方网站(<http://www.eclipse.org/downloads/>)去下载其软件包。Eclipse 无需安装,直接解压缩即可运行。

1.2.4 编写并运行一个显示“Hello World”的 Java 应用程序

熟悉 Eclipse 的开发环境,新建一名为 Project1 的 Java Project,然后在工程中建立一个包含 main 方法的类 Hello,运行并查看结果。

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

实验 2 结构化编程

2.1 实验目的与要求

- (1)掌握 Java 的基本数据类型、运算符和表达式；
- (2)掌握 Java 分支结构、循环结构的程序设计方法；
- (3)学会编写简单的 Java 程序。

2.2 预习内容

Java 的基本数据类型、运算符、表达式和分支、循环语句

2.3 实验内容

2.3.1 基本数据类型

编写一个简单的 Java 程序,定义各种数据类型的变量,代码如下:

```
public class Java2_1{  
    public static void main(String args[]){  
        byte b=0x55;  
        short s=0x1234;  
        int i=0x12345678;  
        long l=0x123456789L;  
        char c='a';  
    }  
}
```

```

float f=3.14F;
double d=3.1415926;
boolean B=true;
String S="Java 程序设计 ";
System.out.println(" 字节型变量 b = "+b);
System.out.println(" 短整型变量 s = "+s);
System.out.println(" 整型变量 i = "+i);
System.out.println(" 长整型变量 l = "+l);
System.out.println(" 字符型变量 c = "+c);
System.out.println(" 浮点型变量 f = "+f);
System.out.println(" 双精度变量 d = "+d);
System.out.println(" 布尔型变量 B = "+B);
System.out.println(" 字符串类对象 S = "+S);
}
}

```

(1) 编译并运行该程序, 写出运行结果。

(2) 修改上述程序, 用 `byte b=0x552` 替换 `byte b=0x55` 编译并运行程序, 如果有错误, 分析原因。

(3) 修改上述程序, 用 `short s=0x12345` 替换 `short s=0x1234` 编译并运行程序, 如果有错误, 分析原因。

(4) 修改上述程序, 用 `int i=0x123456789` 替换 `int i=0x12345678` 编译并运行程序, 如果有错误, 分析原因。

(5) 修改上述程序, 用 `long l=0x123456789` 替换 `long l=0x123456789L` 编译并运行程序, 如果有错误, 分析原因。

(6) 修改上述程序, 用 `float f=3.14` 替换 `float f=3.14F` 编译并运行程序, 如果有错误, 分析原因。

2.3.2 循环结构

(1) 编写一个 Java 程序, 通过键盘输入一组数据, 求出其平均值并输出。

(2) 编写一个 Java 程序, 输出 100 以内所有的素数, 输出时每行显示 5 个数, 输出结果能满足如下形式:

```

2      3      5      7      11
13     17     19     23     29

```

31	37	41	43	47
53	59	61	67	71
73	79	83	89	97

2.4 实验步骤

2.4.1 基本数据类型

(1)运行结果如下：

```
字节型变量 b = 85
短整型变量 s = 4660
整型变量 i = 305419896
长整型变量 l = 4886718345
字符型变量 c = a
浮点型变量 f = 3.14
双精度变量 d = 3.1415926
布尔型变量 B = true
字符串对象 S = Java程序设计
```

(2)编译出错,提示 byte b=0x552;可能精度损失。原因在于 byte 型变量内存中占 1 个字节,取值范围是 -128~127,0x552 超出其取值范围,不能赋值给一个 byte 型变量。

(3)与 2)类似错误,提示 short s=0x12345;可能精度损失。原因在于 short 型变量内存中占 2 个字节,0x12345 超出其取值范围,不能赋值给一个 short 型变量。

(4)编译出错,提示 0x123456789 过大的整数。原因在于,一个整型常量默认情况下是 int 型,int 型内存中占 4 个字节,0x123456789 超出其取值范围,是一个过大的整数。如果,我们在 0x123456789 后面加后缀 L 是否可以呢?编译出错,提示 int i=0x123456789L;可能精度损失。0x123456789L 超出 int 型变量的取值范围。

(5)编译出错,与 4)相同。

(6)编译出错,提示 float f=3.14;可能损失精度,原因在于一个浮点型常量,默认情况是 double 型的,除非加后缀 f 或 F。所以把一个 double 型常量赋值给一个 float 型变量,存在精度损失

2.4.2 循环结构

(1)提示:一组数据可以通过命令行输入,因为 Java 应用程序的 main 方法中的参数 args 能接受从键盘输入的字符串参数。程序代码参考如下:

```
public class Java2_2 {  
    public static void main(String args[] ) {  
        double n,sum=0,value;  
        for(int i=0;i<args.length;i++) {  
            value=Double.parseDouble(args[i]);  
            sum=sum+value;  
        }  
        n=sum/args.length;  
        System.out.println("average="+n);  
    }  
}
```

编辑 Java2_2.java,假设存放在路径 E:\java 下,则编译并运行如下:

```
E:\java>javac Java2_2.java  
  
E:\java>java Java2_2 "1.2" "3.4" "4.5"  
average=3.0333333333333333
```

分析:

a) 参数的个数可以通过 args.length 来获得;

b) 参数“1.2”,“3.4”,“4.5”分别赋给了 main 方法中的参数 args[0]、args[1]和 args[2];

c) 由于参数都是字符串型的,在程序中通过 Double.parseDouble(string)将字符串转化为 Double 型,再进行运算。

(2) 提示:素数是指除了 1 和自身外,不能被其他自然数整除的数。

程序代码参考如下:

```
public class Java2_3{  
    public static void main(String[] args){  
        int i,j;  
        int n=0;  
        for(i=2;i<=100;i++){  
            for(j=2;j<=i-1;j++){  
                if(i%j==0)  
                    break;  
            }  
            if(j==i){  
                n++;  
                System.out.print(i+"\t ");  
            }  
            if(n==5){  
                System.out.println();  
                n=0;  
            }  
        }  
    }  
}
```

```
        n=0;  
        System.out.println();  
    }  
}  
}
```

运行结果如下：

2	3	5	7	11
13	17	19	23	29
31	37	41	43	47
53	59	61	67	71
73	79	83	89	97

2.5 思考与讨论

- (1) Java 有哪些基本数据类型, 它们各自所占的内存大小是几个字节, 取值范围是多少?
- (2) for 循环与 while 循环各有什么特点, 它们的区别是什么?
- (3) break 和 continue 在流程控制中的作用是什么, 它们有什么区别?