《Java 语言编程高级》实验指导书

院系名称：经管信息学院

课程代码：31090720

总学时数：80

适用专业：大数据技术与应用专业

编 制 人：韦 祥、杨 超

编制日期：2019 年 12 月

审 核 人：

审 定 人：

《Java 语言编程高级》实训指导书

一、实训目的与要求

《Java 语言编程高级》 是计算机科学与技术专业的一门专业核心课， 课程开设的目的就是帮助学生建立面向对象编程的基本思想。通过本课程的学习，使学生在短时间内理解 java 面向对象编程的基本原则， 以及对 java 数据库编程、WEB 服务器编程有一个基本的了解，并为后继课程打下坚实的理论基础和编程基础。

实验课是本门课程的重要环节。实验内容以验证性实验和设计性实验相结合为主，以综合性实验为辅。实验过程中要求使用 JAVA 语言独立进行编程和调试。通过实验主要要达到下述目的：

1、加深对面向对象编程思想的理解。

2、熟悉 JAVA 语言的语言要素， 通过大量的验证性实验， 巩固学生在理论课上所学的知识点。

3、通过大量的实验编写来提高学生对程序的编写、调试、连接、运行全过程，积累程序调试经验。

4、学习如何系统的设计程序、使学生逐渐建立程序设计的系统观，养成良好的编程习惯和编程风格。

二、实训内容

（一）实例实训

学生在实验前必须做好上机的各项准备，按教师的要求进行上机实验。具体要求如下：

1、根据实验指导书提前做好上机预习。程序可以是由教师布置安排的或是自编的。自编程序应书写整齐，并经人工检查无误，以提高上机效率。对程序中自己有疑问的地方，应作出记号，以便在上机时给予注意。

2、 必须携带教材和相关参考资料上机。

3、 调试程序的过程应独立完成。

独立上机是学生独立思维能力独自动手能力的一个重要途径， 上机过程

中出现的问题，一般应由学生独立处理，不要轻易举手问老师或周围同学。对常见的“出错信息”应尽快熟悉其含义，并在总结经验的基础上迅速排出常见的错误。

（二）总结

对学生的全部作品进行考核，并选择典型的案例对实训的结果进行考核。

二、参考课时

标题	实训内容	实训课时
实训一	基本知识回顾	3
实训二	设计模式	3
实训三	常用实用类	9
实训四	输入与输出流	9
实训五	泛型与链表	6
实训六	泛型与散列映射	6
实训七	JDBC 数据库操作	12
实训八	Java 多线程	3
实训九	Java 网络基础	3
实训十	总结	3
总计		60

三、实训材料准备

（一）软件准备

每人配置一台电脑安装 JDK，以及 Blue J、Eclipase。

（二）硬件准备

网络条件：与因特网连接的局域网。

教师用机：Windows 7 及以上版本。

学生用机：Windows 7 及以上版本。

四、综合实训考核办法：

系统文档　20 分

编写代码　40 分

程序调试　10 分

实训出勤　20 分

技术含量　10 分

目　　录

实训一 基本知识回顾

一、 实训目的和要求

程序调试通过运行成功后，应及时做好程序清单和运行结果的记录， 实验结束后及时填写实验报告。实验报告应包括如下内容：

实验时间

实验内容

源程序清单

运行结果及分析结论

实验报告要求记录程序调试中出现的错误提示。 （英文、中文对照），若有没通过的程序，分析原因。

二、 实训内容

Java 语言高级编程内容。

三、 实训准备

一台电脑安装 JDK，以及 Blue J、Eclipase。

四、 实训步骤

各实训指导教师按照所代学生的情况不同选择性地按下列步骤温习 Java 语言高级编程的重点难点知识。

(一) 设计模式

1. 策略模式
2. 访问者模式
3. 适配器模式

(二) 常用实用类

1. 掌握 String 类的常用方法
2. 掌握 Date 类以及 Calendar 类的常用方法管理链接
3. 掌握接 BigInteger 类的常用方法

(三) 输入与输出流

1. 掌握使用输入输出流进行文件的读写操作

(四) 泛型和链表

1. 掌握 ArrayList 类与 LinkedList 类的用法

（五） JDBC 数据库操作

1. 掌握 Java 数据库联接和访问的基本方法
2. 掌握数据库 Jar 包的路径设置
3. 掌握创建和操作数据库；创建连接
4. 掌握通过 JDBC 对数据库中的数据进行增、删、改、查

（六） Java 多线程

1. 掌握如何通过继承 Thread 类实现多线程的创建
2. 掌握 Thread 类中 run()方法和 start()方法的使用
3. 掌握如何通过实现 Runnable 接口方式创建多线程

五、 实训方法

使用投影进行讲解演示，并抽样进行检查。

六、 考核办法

此部分实训内容采用抽样考察的方法，考核以操作的熟练程度和正确性为评分标准，以 A（优秀）、B（良好）、C（及格）、D（不及格）为成绩标准。

七、 思考和练习

在应用程序中，如果要对线程进行调度，最直接的方式就是设置线程的优先级。这时，可以通过线程的 setPriority() 方法来设置线程优先级别，实现对线程的调度功能。为了让初学者掌握线程的优先级，在案例中创建 3 个线程，分别为它们设置不同的优先级来演示不同优先级线程的调度。

实训二 设计模式

一、 实训目的和要求

使用 Java 面向对象编程语言实现几种常用的设计模式，加深对这些模式的理解，包括策略模式、访问者模式、适配器模式。

二、 实训内容

使用 Java 面向对象编程语言实现几种常用的设计模式，加深对这些模式的理解，包括策略模式、 访问者模式、 适配器模式。包括根据实例绘制相应的模式结构图、编写模式实现代码，运行并测试模式实例代码。

三、 实训准备

一台电脑安装 JDK，以及 Blue J、Eclipase。

四、 实训步骤

1. 绘制策略模式结构图并用 Java 面向对象编程语言实现该模式；

问题描述：某商家的洗衣机准备 818 搞活动，进行促销，有几种促销手段：

（1）满 300 减 50

（2）打 8 折

（3）打 9 折基础上，折后价满 200 减 20

现在要画出类图并编写源码，实现输出洗衣机在 818 活动时的价格。

目标： 请用 strategy 模式设计解决方案。方案要能够使得在销售洗衣机（即使是同一种洗衣机）时可以灵活的选用价格计算方法，并且可以很容易地增加或修改价格计算方

法而不至于对整个系统的维护造成困难。

2. 制访问者模式结构图并用 Java 面向对象编程语言实现该模式；

3. 绘制适配器模式结构图并用 Java 面向对象编程语言实现该模式；

各种模式实例的内容可参考教材第 8 章中的设计模式实例，也可根据自己对该设计模式的理解，自行设计模式结构图并用 Java 实现。

五、 实训方法

使用投影讲解演示，并抽样进行检查。

六、 考核办法

此部分实训内容采用抽样考察的方法，考核以操作的熟练程度和正确性为评

分标准，以 A（优秀）、B（良好）、C（及格）、D（不及格）为成绩标准。

七、 思考和练习

1.绘制装饰模式结构图并用 Java 面向对象编程语言实现该模式；

2.绘制外观模式结构图并用 Java 面向对象编程语言实现该模式；

3.绘制模板方法模式结构图并用 Java 面向对象编程语言实现该模式；

实训三 常用实用类

一、 实训目的和要求

1、掌握 String 类的常用方法

2、掌握 Date 类以及 Calendar 类的常用方法

3、掌握接 BigInteger 类的常用方法

4、掌握怎样使用 Pattern 类和 Match 类检索字符串

二、 实训内容

根据实验步骤里的源代码，按照注释要求，完成代码填空，使程序能够运行得出结果。

1) 实 验 2-1 检索图书

2) 实 验 2-2 购物小票

3) 实 验 2-3 比较日期

4) 实 验 2-4 处理大整数

5) 实 验 2-5 替 换 IP

6）实验 2-6 String 类的常用方法。

三、 实训准备

一台电脑安装 JDK，以及 Blue J、Eclipase。

四、 实训步骤

实验 2-1 检索图书模板代码

FindMess.java

```
public class FindMess {
public static void main(String[] args) {
String mess="书名： Java 程序设计，出版社时间： 2011.10.01， "+"
出版社：清华大学出版社，价格： 29.8 元，页数： 389 页";
if(【代码 1】){// 判断 mess 中是否含有 " 程序"System.out.println("
图书信息包含有 \" 程序 \"");
}
int index=// 【代码 2】//mess 调用 indexOf(String s) 返回 mess 中第
```

2 个冒号的位置

```
String date=mess.substring(index+1,index+11);
System.out.println(date);
int pricePosition=// 【代码 3】//mess 调用 indexOf(String s) 返回首次出现 "价格" 的位置 int endPosition=mess.indexOf(" 元");
String priceMess=mess.substring(pricePosition+3,endPosition);
System.out.println(" 图 书 价 格 "+priceMess); double price=Double.parseDouble(priceMess); if(price>=29){
System.out.println(" 图书价格 "+price+" 大于或等于 29 元");
}else{
System.out.println(" 图书价格 "+price+" 小于 29 元");
}
index=// 【代码 4】//mess 调用 laseIndexOf(String s,int start) 返回最后 1 个冒号的位置 endPosition=mess.lastIndexOf(" 页");
String pageMess=mess.substring(index+1,endPosition); int p=Integer.parseInt(pageMess);
if(p>=360){
System.out.println(" 图书价格 "+p+" 大于或等于 360");

}else{
}
}
}

System.out.println(" 图书价格 "+p+" 小于 360");
```

实验 2-2 购物小票

模板代码

FoundPrice.java

```
import  java.util.regex.Matcher;  import  java.util.regex.Pattern;
public class FoundPrice {
public static void main(String[] args) {
String s=" 牛奶:89.8 元，香肠 :12.9 元，啤酒 :69 元，巧克力 :132 元
"; String regex=// 【代码  1】// 匹配数字与小数点的正则表达式的字
符串 int number=0;
double sum=0.0;
Pattern p = //【代码  2】//使用 regex 初始化模式对象 pattern
Matcher m = p.matcher(s);
while(m.find()) {
System.out.println(m.group()); number=number+1;
try{
sum=sum+//【代码  3】//获取子匹配成功结果字符串并转成  Double  型
类
}
catch(NumberFormatException
e){ System.out.println(e.getMessage());
}
}

System.out.println(" 购物小票中的商品种类：  "+number+"  种  ");
System.out.println(" 购物小票中的价格总额：  "+sum+" 元");
}
}
```

实验 2-3 比较日期模板代码

CompareDate.java

```
import java.util.*;
public class CompareDate{
public static void main(String args[ ]) {
Scanner scanner=new Scanner(System.in); System.out.println(" 输入第一个年 ,月,日数据 "); System.out.print(" 输入年份 ");
short yearOne=scanner.nextShort();
System.out.print(" 输入月份 ");
byte monthOne=scanner.nextByte();
System.out.print(" 输 入 日 份 "); byte dayOne=scanner.nextByte();
System.out.println(" 输 入 第 二 个 年 , 月 , 日 数 据 ");
System.out.print(" 输入年份 ");
short yearTwo=scanner.nextShort(); System.out.print(" 输入月份 ");
byte monthTwo=scanner.nextByte();
System.out.print(" 输 入 日 份 "); byte dayTwo=scanner.nextByte();
Calendar calendar=//【代码 1】// 初始化日历对象
// 【代码 2】// 将 calendar 的时间设置为 yearOne 年 monthOne 月 dayOne 日 long timeOne=// 【代码 3】 //calendar 表示的时间转换成毫秒 calendar.set(yearTwo,monthTwo-1,dayTwo);
long timeTwo=calendar.getTimeInMillis();
Date date1=//【代码 4】 // 用 timeOne 做参数构造 date1 Date date2=new Date(timeTwo);
if(date2.equals(date1))
{
System.out.println(" 两个日期的年、月、日完全相同 ");
}
else if(date2.after(date1))
{
```

```
System.out.println(" 您输入的第二个日期大于第一个日期    ");
}
else if(date2.before(date1))
{
System.out.println(" 您输入的第二个日期小于第一个日期    ");
}
long days=// 【代码 5】//使用 timeOne,timeTwo 计算两个日期相隔天数
System.out.println(yearOne+" 年"+monthOne+" 月"+dayOne+" 日和"
+yearTwo+" 年"+monthTwo+" 月"+dayTwo+" 相隔"+days+" 天");
}
}
```

实验 2-4 处理大整数

模板代码

```
HandleBigInteger.java
import java.math.*;
public class HandleBigInteger {
public static void main(String args[]){
BigInteger n1=new BigInteger("98765432198765432198765432l"), n2=new
BigInteger("123456789123456789123456789"),
result=null;
result=// 【代码 1】//n1 和 n2 做加法运算 System.out.println("
和:"+result.toString()); result=// 【代码 2】//n1 和 n2 做减法运算
System.out.println(" 差:"+result.toString()); result=// 【代码 3】
//n1 和 n2 做乘法运算 System.out.println(" 积:"+result.toString());
result=// 【代码 4】//n1 和 n2 做除法运算 System.out.println("
商:"+result.toString()); BigInteger m=new BigInteger("1968957"),
```

```
COUNT=new BigInteger("0"), ONE=new BigInteger("1"), TWO=new BigInteger("2");
System.out.println(m.toString()+" 的 因 子 有 :"); for(BigInteger i=TWO;i.compareTo(m)<0;i=i.add(ONE)){
if((n1.remainder(i).compareTo(BigInteger.ZERO))==0){ COUNT=COUNT.add(ONE);
System.out.print("  "+i.toString());

}
}

System.out.println("");
System.out.println(m.toString()+" 一共有 "+COUNT.toString()+" 个因子 ");
}
}
```

实验 2-5 替 换 IP

模板代码

ReplaceErrorWord.java

```
import java.util.regex.Matcher; import java.util.regex.Pattern;
public class ReplaceErrorWord {
public static void main(String[] args) {
String str=" 登录网站 :222.128.89.253"; Pattern pattern;

Matcher matcher;
String regex="[\\d]{1,3}[.][\\d]{1,3}[.][\\d]{1,3}[.][\\d]{1,3}";
```

```
pattern=//【代码  1】// 使用 regex 初始化模式对象        pattern
matcher=//【代码  2】//得到检索  str 的匹配对象  matcher
String IP="";
while(matcher.find()){ IP=matcher.group();
System.out.print(matcher.start()+" 位置出现： ");
System.out.println(IP);

}
System.out.printf(" 将 %s  替 换 为  202.192.78.56\n",IP); String
result=matcher.replaceAll("202.192.78.56");
System.out.println(result);
}
}
```

实验 2-6 String 类的常用方法模板代码

StringExample.java

```
class StringExample
{   public static void main(String args[])
{   String s1=new String("you are a student"), s2=new String("how are
you");
if( 【代码 1】) // 使用 equals 方法判断 s1 与 s2 是否相同
{

System.out.println("s1 与 s2 相同");
}
else
{
```

```
System.out.println("s1 与 s2 不 相 同 ");
}

String s3=new String("22030219851022024");
if( 【代码  2】) // 判断  s3 的前缀是否是“  220302”。
{
System.out.println(" 吉林省的身份证    ");
}

String s4=new String(" 你"), s5=new String(" 我");
if( 【代码  3】)// 按着字典序  s4 大于 s5 的表达式。
{

System.out.println(" 按字典序 s4 大于 s5");
}
else

{
System.out.println(" 按字典序 s4 小于 s5");
}

int position=0;
String path="c:\\java\\jsp\\A.java";
position= 【代码 5】 //获取 path 中最后出现目录分隔符号的位置
System.out.println("c:\\java\\jsp\\A.java 中最后出现的          位
置 :"+position);String fileName=【代码 6】//获取 path 中“ A.java ”
子字符串。System.out.println("c:\\java\\jsp\\A.java 中含有的文件
名 :"+fileName); String s6=new String("100"),
s7=new String("123.678");
```

```
int n1= 【代码 7】 //将 s6 转化成 int 型数据。double n2= 【代码 8】 //将 s7 转化成 double 型数据。double m=n1+n2;
System.out.println(m);
String s8=【代码 9】 //String 调用 valuOf(int n) 方法将 m 转化为字符串对象 position=s8.indexOf(".");
String temp=s8.substring(position+1);
System.out.println(" 数字"+m+" 有"+temp.length()+" 位小数 ") ;
String s9=new String("ABCDEF");
char a[]=【代码 10】 //将 s8 存放到数组 a 中 。 for(int i=a.length-1;i>=0;i--)
{

System.out.print(" "+a[i]);
}
}
}。
```

五、 实训方法

在机房进行实训，由实训指导教师亲临指导。

六、 考核办法

此部分实训内容采用全体考察的方法，考核以操作的熟练程度和正确性为评分标准，以 A（优秀）、B（良好）、C（及格）、D（不及格）为成绩标准。

七、 思考和练习

计算某年、某月、某日和某年、某月、某日之间的天数间隔。要求年、月、日通过 main 方法的参数传递到程序中。

实训四 输入与输出流

一、 实训目的和要求

掌握使用输入输出流进行文件的读写操作。

二、 实训内容

根据实验步骤里的源代码，按照注释要求，完成代码填空，使程序能够运行得出结果。

1)实验 3-1 实现文件加密

实验 3-2 给文件的内容添加行号。

三、 实训准备

一台电脑安装 JDK，以及 Blue J、Eclipase。

四、 实训步骤

实验 3-1 实现文件加密

下面程序实现功能：将已存在的文本文件加密后存入另一个文本文件中。请将【代码 1】～

【代码 6】替换为 Java 程序代码。SecretExample.java

```
import java.io.*;
public class SecretExample
{
public static void main(String args[ ])
{
File fileOne=new File("hello.txt"), fileTwo=new File("hello.secret");
char b[]=new char[100];
try{
FileReader in= 【代码 1】// 创建指向 fileOne 的字符输入流
FileWriter out= 【代码 2】 // 创建指向 fileTwo 字符输出流
int n=-1;
while((n=in.read(b))!=-1)
{
```

```
for(int i=0;i<n;i++)
{
b[i]=(char)(b[i]^'a');
}
【代码 3】  // out 将数组 b 的前 n 单元写到文件
}
【代码 3】  // out 关 闭
in= 【代码 4】// 创建指向 fileTwo 的 字 符 输 入 流
System.out.println(" 加密后的文件内容 :");
while((n=in.read(b))!=-1)
{

String str=new String(b,0,n); System.out.println(str);
}
in= 【代码 5】// 创建指向 fileTwo 的 字 符 输 入 流
System.out.println(" 解密后的文件内容 :");
while((n=in.read(b))!=-1)
{
for(int i=0;i<n;i++)
{
b[i]=(char)(b[i]^'a');
}
System.out.printf(new String(b,0,n));
}
【代码 6】  // in 关 闭
}
catch(IOException e)
{
System.out.println(e);
```

```
}
}
}
```

实验 3-2 给文件的内容添加行号

下面程序实现功能：给文件的内容添加行号。请将【代码 1】～【代码 14】替换为 Java 程序代码。

```
ReadExample.java import java.io.*;
public class ReadExample
{
public static void main(String args[ ])
{
File file=new File("c:/1000","hello.txt"); File tempFile=new File("temp.text"); try{
FileReader inOne= 【代码 1】 // 创建指向文件 file 的输入流 BufferedReader inTwo= 【代码 2】// 创建指向 inOne file 的输入流 FileWriter tofile= 【代码 3】// 创建指向文件 tempFile 的输出流 BufferedWriter out= 【代码 4】// 创建指向 tofile 的输出 流 String s=null;
int i=0;
s=【代码 5】 // inTwo 读取一行
while(s!=null)
{
i++;
out.write(i+" "+s);

out.newLine();

s=【代码 6】
```

```
}       // inTwo  读取一行
inOne.close();
inTwo.close();
out.flush();
out.close();
tofile.close();
inOne= 【代码 7】inTwo= 【代码 8】tofile= 【代码 9】
out= 【代码 10】 // 创建指向文件  tempFile 的输入流
//  创建指向  inOne file 的输入流
//  创建指向文件  file 的输出流
//  创建指向  tofile 的输出流
while((s= 【代码  11】)!=null) // inTwo 读取一行
{
out.write(s); out.newLine();
}
inOne.close(); inTwo.close(); out.flush();
out.close(); tofile.close();
inOne= 【代码  12】  //  创建指向文件  file 的输入流
inTwo= 【代码  13】  // 创建指向  inOne file 的输入流
while((s= 【代码  14】 )!=null) // inTwo  读取一行
{
System.out.println(s);
}
inOne.close(); inTwo.close(); tempFile.delete();
}
catch(IOException e)
{
System.out.println(e);
}
```

}

}。

五、 实训方法

首先由实训指导教师在机房讲解站点规划书的书写方法和基本格式，其次由学生针对所选的主题进行站点规划。

六、 考核办法

此部分实训内容采用全体考察的方法，考核以站点规划书的完整性、实用性和创造性为评分标准，以 A（优秀）、B（良好）、C（及格）、D（不及格）为成绩标准。

七、 思考和练习

1.StringBuffer 类有很多操作字符的方法， 其中 append()和 insert()是常用的添加字符方法， delete()是常用的删除字符方法。为了让初学者对 StringBuffer 类中的添加、删除方法有

更深入的了解，本案例将演示 StringBuffer 对象的添加、删除操作。

```
public class Example06 {
// 创建对象
StringBuffer sb = new StringBuffer(); System.out.println("sb:" +
sb);
System.out.println("****StringBuffer 的添加方法 ****");
sb.append(100).append("hello").append(true).append(12.5);
System.out.println("使用 StringBuffer 对象添加任意数据类型的
数据


后 sb:" + sb);sb);
}
sb.insert(8, "world");
System.out.println("在 sb 的第 9 个位置插入 'world' 字符串
后 sb:" +System.out.println("****StringBuffer 的删除方法 ****");
```

sb.deleteCharAt(1);System.out.println(" 删除 sb 中第 2 个位置的字符后 sb:" + sb); sb.delete(5, 10);System.out.println(" 删除 sb 中第 6 个到第 11 个字符后的 sb:" + sb);

}

2、String 类中提供了许多对字符串进行替换、切割操作的方法，其中每个方法都有其

各自的作用， 为了让初学者能快速熟悉替换、 切割方法的用法， 本案例将针对不同需求使用不同的替换、切割方法，并根据输出结果进行参照学习。

'p'));

public class Example05 {

public static void main(String[] args) { String s = "helloworld";

System.out.println("将字符串 s 中的字符 l 替 换 成 p 后 ："+s.replace('l',

System.out.println("将字符串 s 中的字符 ll 替换成 ak47 后："+s.replace("ll", "ak47"));

String ages = "20-30";

String[] strArray = ages.split("-");

for (int x = 0; x < strArray.length; x++) {

System.out.println("strArray 数组中的索引为 "+x+" 处的值是："

+strArray[x]);

}

String name = " admin hello ";

System.out.println("去掉首尾空格后的字符串 name ： " + name.trim()); String s1 = "hello";

String s2 = "aello";

```
        System.out.println("按照默认字典的顺序比较字符串    s1 和 s2 ： "+
s1.compareTo(s2));// 7
    }
    }
```

实训五 泛型和链表

一、 实训目的和要求

掌握 ArrayList 类与 LinkedList 类的用法。

二、 实训内容

根据实验步骤里的题目要求与实验结果截图，完成各个类的源代码。

三、 实训准备

一台电脑安装 JDK，以及 Blue J、Eclipase。

四、 实训步骤

有四个类，主类 Store 在包 cn.edu.nwsuaf.jp.p4 中， Mobile 、Mp3Player 、Product 在包 cn.edu.nwsuaf.jp.p4.data 中，Mobile 、Mp3Player 是 Product 的子类， Product 类实现 Comparable 接口，重写了 Comparable 接口中方法 compareTo，实现了 product 对象按照价格排序。

基本要求：

（ 1）写出 Product 、Mobile 、Mp3Player 类的源码，是否有构造函数根据输出结果自行决定。

（ 2）在主类 Store 中实例化多个 Mobile 类与 Mp3Player 的实例，分别将这些实例用

ArrayList 与 LinkedList 存放，最后用 StringBuiler 存放并将其输出。

（ 3）用迭代器（ iterator ）将实例对象输出（要求用加强型 for 循环）。

输出结果如下所示：

```
用StringBuiler输出:
E365 on china Mobile, 1780.0 RMB
E3330 on china Mobile, 1450.0 RMB
Xlive XM Mp3Player(256MB), 930.0 RMB
Meiz0 E5 (512MB), 580.0 RMB
Meiz0 X3 (256MB), 399.0 RMB

用ArrayList或LinkedList迭代器输出:
E365 on china Mobile, 1780.0 RMB
E3330 on china Mobile, 1450.0 RMB
Xlive XM Mp3Player(256MB), 930.0 RMB
Meiz0 E5 (512MB), 580.0 RMB
Meiz0 X3 (256MB), 399.0 RMB
```

五、 实训方法

首先由实训指导教师在机房讲解站点规划书的书写方法和基本格式，其次由学生针对所选的主题进行站点规划。

六、 考核办法

此部分实训内容采用全体考察的方法，考核以操作的熟练程度和正确性为评分标准，以 A（优秀）、B（良好）、C（及格）、D（不及格）为成绩标准。

七、 思考和练习

List 集合遍历

在程序开发中，经常需要遍历集合中的所有元素。针对这种需求， JDK 专门提供了一个接口 Iterator 。为了使初学者熟悉 Iterator 迭代器的使用，本案例将演示如何使用 Iterator 迭代器遍历集合中的元素。设计思路（实现原理）

实训六 泛型和散列映射

一、 实训目的和要求

掌握 Map 接口及其实现类的用法。

二、 实训内容

根据实验步骤里的题目要求，完成各个类的源代码。

三、 实训准备

Dreamweaver 8 中文版 、WORD2000 以上版本、Photoshop7.0 以上版本、 与因特网连接的局域网。

四、 实训步骤

已知某学校的教学课程内容安排如下：

老师	课程
Tom	CoreJava
John	Oracle
Susan	Oracle
Jerry	JDBC
Jim	Unix
Kevin	JSP
Lucy	JSP

完成下列要求：

（1） 使用一个 Map，以老师名字作为键，以老师的授课的课程名作为值，表示上述课程安排。

（2） 增加了一位新老师 Allen 教 JDBC

（3） Lucy 改为教 CoreJava

（4） 用两种方式遍历 Map ，输出所有的老师及老师教授的课程

（5） 利用 Map ，输出所有教 JSP 的老师

五、 实训方法

机房上网并利用本机软件完成。

六、 考核办法

此部分实训内容采用全体考察的方法，考核以操作的熟练程度和正确性为评

分标准，以 A（优秀）、B（良好）、C（及格）、D（不及格）为成绩标准。

七、 思考与练习

操作 HashMap 集合的方法设计思路（实现原理）

（1） 编写类 Example13 ，创建一个 HashMap 集合，并向集合中添加元素

（2）使用一系列方法分别对定义的集合进行增加、移除、判断等操作

（3）运行程序，根据输出结果，分析这些方法对 HashMap 集合的操作

```
mport java.util.*;
public class Example13 {
public static void main(String[] args) {
// 创建集合对象
Map map = new HashMap();
// 添加元素
map.put("czbk001", "林青霞
map.put("czbk003", ");
map.put("czbk002", "林志颖 ");
// 添加元素，如果键存在，就替换，返回以前与  key 关联的值。
System.out.println(map.put("czbk003", "  林志炫 "));
// V remove(Object key) ：根据指定的键删除键值对。
System.out.println("remove:" + map.remove("czbk003"));
// 判断指定的键是否在集合中存在
System.out.println("containsKey:" + map.containsKey("czbk002"));
System.out.println("containsKey:" + map.containsKey("czbk007"));
// 判断指定的值是否在集合中存在
System.out.println("containsValue:"  + map.containsValue("  林志颖 "));
// 判断集合是否为空
System.out.println("isEmpty:" + map.isEmpty());
// 集合中元素的个数
System.out.println("size:" + map.size());
```

```
// 输出集合对象名称
System.out.println("map:" + map);
}
}
```

运行结果如图所示：

实训七 JDBC 数据库操作

一、 实训目的和要求

1、掌握 Java 数据库联接和访问的基本方法

2、掌握数据库 Jar 包的路径设置；

3、掌握创建和操作数据库；创建连接；

4、掌握通过 JDBC 对数据库中的数据进行增、删、改、查。

二、 实训内容

根据 JDBC 连接数据库的程序包含的 7 个步骤，以 MySQL 为数据库操作平台，编写 Java 连接数据库并操作数据的实例。

三、 实训准备

一台电脑安装 JDK，以及 Blue J、Eclipase。

四、 实训步骤

创建一个以 JDBC 连接数据库的程序，包含 7 个步骤：

1、加载 JDBC 驱动程序：

在连接数据库之前， 首先要加载想要连接的数据库的驱动到 JVM（ Java 虚拟机）， 这通过 java.lang.Class 类的静态方法 forName(String className) 实现。

例如：

```
try{
//加载 MySql 的驱动类
Class.forName("com.mysql.jdbc.Driver") ;
}catch(ClassNotFoundException e){
System.out.println(" 找不到驱动程序类 ，加载驱动失败！ ");
e.printStackTrace() ;
}
```

成功加载后，会将 Driver 类的实例注册到 DriverManager 类中。

2、提供 JDBC 连接的 URL 、 username、 password

例如：

```
//连接 MySql 数据库，用户名和密码都是 root String url =
```

"jdbc:mysql://localhost:3306/test" ; String username = "root" ; String password = "root" ;

3、创建数据库的连接

要连接数据库，需要向 java.sql.DriverManager 请求并获得 Connection 对象， 该对象就代表一个数据库的连接。

使用 DriverManager 的 getConnectin(String url , String username , String password) 方法传入指定的欲连接的数据库的路径、数据库的用户名和 密码来获得。

例如：

```
try{
Connection con = DriverManager.getConnection(url , username , password ) ;
}catch(SQLException se){
System.out.println(" 数据库连接失败！ "); se.printStackTrace() ;
}
```

4、创建一个 Statement

要执行 SQL 语句，必须获得 java.sql.Statement 实例，Statement 实例分为以下 3 种类型：

（ 1）执行静态 SQL 语句。通常通过 Statement 实例实现。

（ 2）执行动态 SQL 语句。通常通过 PreparedStatement 实例实现。

（ 3）执行数据库存储过程。通常通过 CallableStatement 实例实现。具体的实现方式：

```
Statement stmt = con.createStatement() ; PreparedStatement pstmt = con.prepareStatement(sql) ;
CallableStatement cstmt = con.prepareCall("{CALL demoSp(? , ?)}") ;
```

5、执行 SQL 语句

Statement 接口提供了三种执行 SQL 语句的方法： executeQuery 、 executeUpdate 和

execute

（ 1）ResultSet executeQuery(String sqlString) ：执行查询数据库的 SQL 语句 ，返回一个结果集（ ResultSet）对象。

（ 2）int executeUpdate(String sqlString) ：用于执行 INSERT 、 UPDATE 或 DELETE 语 句

以及 SQL DDL 语句，如： CREATE TABLE 和 DROP TABLE 等

（ 3） execute(sqlString): 用于执行返回多个结果集、 多个更新计数或二者组合的 语句。具体实现的代码：

ResultSet rs = stmt.executeQuery("SELECT * FROM ...") ; int rows = stmt.executeUpdate("INSERT INTO ...") ; boolean flag = stmt.execute(String sql) ;

6、处理结果

两种情况：

（ 1）执行更新返回的是本次操作影响到的记录数。

（ 2）执行查询返回的结果是一个 ResultSet 对象。

ResultSet 包含符合 SQL 语句中条件的所有行，并且它通过一套 get 方法提供了对这些 行中数据的访问。

使用结果集（ ResultSet)对象的访问方法获取数据：while(rs.next()){

String name = rs.getString("name") ;

String pass = rs.getString(1) ; // 此方法比较高效

}

7、关闭 JDBC 对 象

操作完成以后要把所有使用的 JDBC 对象全都关闭，以释放 JDBC 资源，关闭顺序和声 明顺序相反：

（ 1）关闭记录集

（ 2）关闭声明

（ 3）关闭连接对象

if(rs != null){ // 关闭记录集

try{

```
}catch(SQLException e){
}
}
if(stmt != null){ // 关闭声
try{
stmt.close() ;
}catch(SQLException e){ e.printStackTrace() ;
}
}
if(conn != null){ // 关闭连接对象
try{
conn.close() ;
}catch(SQLException e){ e.printStackTrace() ;
}
}
```

根据以上步骤，在 MySQL 中创建数据库 mydb ，完成如下任务：

（ 1）在下面空白处写出表结构

（ 2）以增、删、改、查中的任意一种或几种操作为例，首先通过注释写出操作目的，然后给出完整的通过 JDBC 访问数据库的源码。

五、 实训方法

机房利用本机软件完成。

六、 考核办法

此部分实训内容采用抽样考察的方法，

实训八 Java 多线程

一、 实训目的和要求

1.掌握如何通过继承 Thread 类实现多线程的创建。

2.掌握 Thread 类中 run()方法和 start()方法的使用。

3.掌握如何通过实现 Runnable 接口方式创建多线程。

4.掌握如何使用 Thread 类的有参构造方法创建 Thread 对象。

二、 实训内容

1.通过继承 Thread 类方式创建线程，并实现多线程分别打印 0~99 的数字的功能。

2.通过实现 Runnable 接口方式创建线程，并实现多线程分别打印 0~99 的数字的功能。

三、 实训准备

一台电脑安装 JDK，以及 Blue J、Eclipase 。

四、 实训步骤

实验 7-1 继承 Thread 类方式创建线程

在程序开发中，会遇到一个功能需要多个线程同时执行才能完成的情况。

这时，可以通过继承线程类 Thread，并重写 Thread 类中的 run()方法来实现。为了让初学者熟悉如何创建多线程， 在案例中将通过继承 Thread 类方式创建线程， 并实现多线程分别打印 0~99 的数字的功能。

设计思路（实现原理）

1)自定义一个类 Demo ，使其继承 Thread 类。

2)在 Demo 类中重写 run() 方法，在 run() 方法内编写一个 for 循环，循环体内打印： “Demo ：”+当前循环次数。

3)编写测试类 Example01，在 Example01 类的 main() 方法中，创建一个 Demo 对象，并执行其 start()方法，接着编写一个 for 循环，循环体内打印：“ main:” +当前循环次数。

五、 实训方法

使用投影讲解演示，并抽样进行检查。

六、 考核办法

此部分实训内容采用抽样考察的方法，考核以操作的熟练程度和正确性为评分标准，以 A（优秀）、B（良好）、C（及格）、D（不及格）为成绩标准。

七、 思考与练习

优先级线程的调度。

设计思路（实现原理）

1)自定义一个类 Demo，使其实现 Runnable 接口。

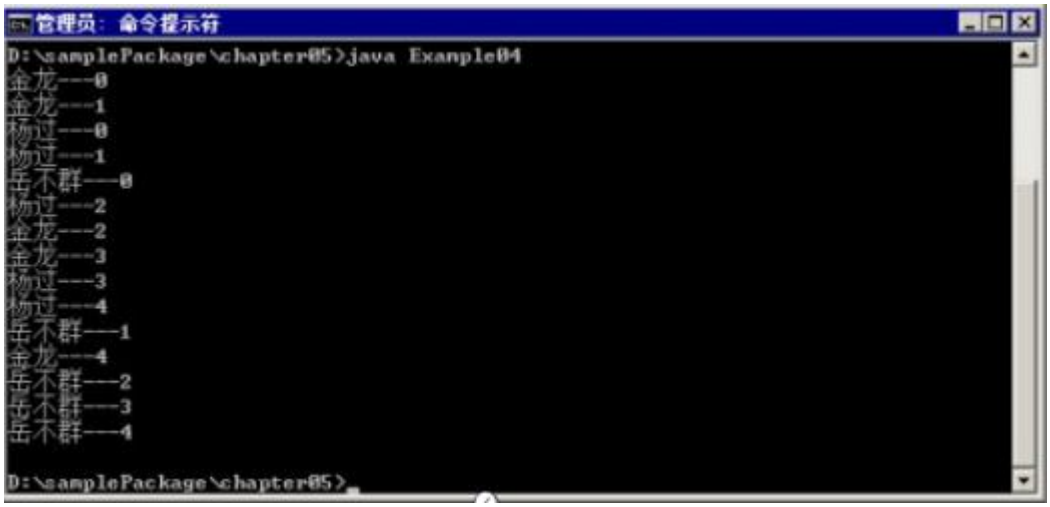
2)在 Demo 类中重写 run() 方法，在方法内编写一个 for 循环，循环体内打印：线程名称+循环次数。

3)编写测试类 Example04，在 Example04 类的 main()方法中，创建一个 Demo 对象，利用 Thread 的构造方法创建三个线程对象并命名，使用 setPriority() 方法将其中两个线程的优先级设为最大和最小，最后开启三个线程的 start()方法。

```
class Demo implements Runnable { public void run() {
for (int x = 0; x < 5; x++) {
System.out.println(Thread.currentThread().getName() + "---" +
x);
}
}
}
public class Example04 {
public static void main(String[] args) { Demo d = new Demo();
Thread t1 = new Thread(d," 杨过");
Thread t2 = new Thread(d," 岳不群");
Thread t3 = new Thread(d," 金龙");
// 设置线程的优先级
t3.setPriority(Thread.MAX_PRIORITY); t2.setPriority(1);
t1.start();
t2.start();
t3.start();
```

```
    }
    }
```

运行结果如下图：


管理员: 命令提示符
D:\samplePackage\chapter05>java Example04
金龙---0
金龙---1
杨过---0
杨过---1
岳不群---0
杨过---2
金龙---2
金龙---3
杨过---3
杨过---4
岳不群---1
金龙---4
岳不群---2
岳不群---3
岳不群---4
D:\samplePackage\chapter05>

实训九 Java 网络基础

一、 实训目的和要求

1.掌握 InetAddress 类的相关 API；

2.掌握如何使用 InetAddress 类中的方法获取计算机的主机名和 IP 地址等信息；

3.通过编写简单的 TCP 程序，掌握 ServerSocket、Socket 类的具体用法。

二、 实训内容

根据实验步骤里的源代码，完成下列实验实验

8-1 继承 Thread 类方式创建线程实验

8-2 继承 Thread 类方式创建线程。

三、 实训准备

一台电脑安装 JDK，以及 Blue J、Eclipase。

四、 实训步骤

实验 8-1 继承 Thread 类方式创建线程

InetAddress 类中提供了一系列与 IP 地址相关的方法，利用这些方法可以获取到指定计

算机的主机名、IP 地址以及连接状态等信息。为了让初学者掌握 InetAddress 类中常用方法的使用，本案例将针对如何通过 InetAddress 类中的方法获取计算机的 IP 地址、主机名等功能进行演示。

设计思路（实现原理）

1)编 写 Example01 类 。

2)在 main()方法中，通过 InetAddress 类的静态方法 getLocalHost()创建实例对象，并通过该对象完成获取计算机主机名和计算机 IP 地址的操作。分别将获取到的主机名和 IP 地址输出。

实验 8-1 继承 Thread 类方式创建线程

InetAddress 类中提供了一系列与 IP 地址相关的方法，利用这些方法可以获取到指定计

算机的主机名、IP 地址以及连接状态等信息。为了让初学者掌握 InetAddress 类中常用方法的使用，本案例将针对如何通过 InetAddress 类中的方法获取计

算机的 IP 地址、主机名等功能进行演示。

设计思路（实现原理）

1)编 写 Example01 类 。

2)在 main()方法中，通过 InetAddress 类的静态方法 getLocalHost()创建实例对象，并通过该对象完成获取计算机主机名和计算机 IP 地址的操作。

分别将获取到的主机名和 IP 地址输出。

```
import java.net.InetAddress;
import java.net.UnknownHostException; public class Example01 {
public static void main(String[] args) throws UnknownHostException {
//  获取本机的    IP 地 址
InetAddress address = InetAddress.getLocalHost();
//  以字符串形式返回 IP 地 址
String ip = address.getHostAddress();
// 获取此 IP 地址的主机名
String name = address.getHostName();
System.out.println("本机的  ip  地址是：）
System.out.println("本机的 hostName 是：）
}
}
```

运行结果如图所示：

五、 实训方法

机房利用本机软件完成。

六、 考核办法

此部分实训内容采用抽样考察的方法，考核以操作的熟练程度和正确性为评

分标准，以 A（优秀）、B（良好）、C（及格）、D（不及格）为成绩标准。

七、 思考与练习

目标:掌握如何使用 TCP 协议完成文件的网络传输功能。

由于 TCP 网络程序能够保证传输数据的完整性和安全性，所以大部分的服务器都会采

用 TCP 协议来实现文件上传的功能。为了让初学者掌握如何使用 TCP 协议完成文件上传功能，本案例将通过使用 TCP 网络协议，实现图片上传的功能。

设计思路（实现原理）

1)编写服务器类 PicUploadServer ，该类包含一个 Socket 类型的私有属性，并提供了该属性的有参构造方法。PicUploadServer 类实现 Runnable 接口，重写 run()方法，在 run()方法内读取客户端上传的图片，并将图片存入服务器指定文件夹中。

2)编写客户端类 PicUpLoadClient ，该类实现了读取指定图片，向指定端口发送图片数据的功能。

3)编写测试类 Example03, 在 main()方法中，通过指定的端口号创建 ServerSocket 对象，并编写死循环，在死循环中，通过 ServerSocket 对象获取 Socket 对象，并开启线程服务。

实训十　总结

一、 实训目的和要求

将学生制作的作品进行综合的考核，并进行总结。

二、 实训内容

1. 对学生作品进行考核。
2. 选择典型的（优秀的和劣质的）作品分别进行总结。

三、 实训准备

一台电脑安装 JDK，以及 Blue J、Eclipase。

连接因特网的局域网。

四、 实训步骤

1. 对学生的作品依次进行综合考核。
2. 抽取典型（优秀和劣质）的作品进行全面的解析。

五、 实训方法

机房利用本机软件完成。

六、 考核办法

此部分实训以考核为主，对学生组品进行整体的考核。采用全体考察的方法，以百分制为满分，具体评分标准如下：

a) 系统文档　20 分
b) 编写代码　40 分
c) 程序调试　10 分
d) 实训出勤　20 分
e) 技术含量　10 分

七、 思考与练习

无。