

《高级语言程序设计》实验指导书

第一部分 《Java 程序设计》

计算机学院软件系

指导老师：汤小春 陈建全

最终改版日：2015/3/18

目录

| | | |
|------------------------|----|----------|
| 实验一 熟悉 Java 程序的开发..... | 2 | 删除的内容: 1 |
| 一、实验目的..... | 2 | 删除的内容: 1 |
| 二、实验内容..... | 2 | |
| 实验二 面向对象的编程技术..... | 4 | |
| 一、实验目的..... | 4 | |
| 二、实验内容..... | 5 | 删除的内容: 4 |
| 1. 基本指导..... | 5 | 删除的内容: 4 |
| 实验三 类的方法 | 7 | 删除的内容: 6 |
| 一、实验目的..... | 7 | 删除的内容: 6 |
| 二、实验内容..... | 7 | 删除的内容: 6 |
| 实验四 类的重用 | 12 | 删除的内容: 9 |
| 一、实验目的..... | 12 | 删除的内容: 9 |
| 二、实验内容..... | 12 | 删除的内容: 9 |
| 实验五 包、接口、类库..... | 16 | 删除的内容: 1 |
| 一、实验目的..... | 16 | 删除的内容: 1 |
| 二、实验内容..... | 16 | 删除的内容: 1 |
| 实验六 异常处理 | 21 | 删除的内容: 1 |
| 一、实验目的..... | 21 | 删除的内容: 1 |
| 二、实验内容..... | 21 | 删除的内容: 1 |

实验一 熟悉 Java 程序的开发

一、实验目的

- (1) 学习使用 JDK 开发工具开发 Java 应用程序;
- (2) 掌握 Java Application 程序的开发过程;
- (3) 掌握 Java Applet 程序的开发过程。

二、实验内容

上机前的重要提示:

- Java 源代码可在任何文本编辑器中输入,但这里建议使用记事本。自实验三开始,可使用 eclipse 环境进行。
- 所有的 Java 源代码都应具有扩展名 “.java”
- 在包含主类的文件中,文件名应与主类的名称相同,并注意有大小写之分。
- 每实验完成时间为 1 学时。
- 每次实验后把实验结果提交给辅导老师。(含实验步骤中的问题回答和 Java 源代码,不要 class 文件.分析和讨论部分可不必提交)
-

1. 基本指导

指导内容 1:

编写并运行第一个 Java Application 程序

实验步骤:

- (1) 开机后,在 java 实验目录下创建 test 子目录。本阶段的 Java 源程序、编译后的字节码文件都放在这个目录中。
- (2) 打开一个纯文本编辑器(如记事本),键入如下程序(注意大小写):

```
import java.io.*;

public class MyClass
{
    private int val1,val2;
    public void myFun(int x,int y)
    {
        val1=x;
        val2=y;
    }
}
```

```

        System.out.println("the sum is "+(val1+val2));
    }

    public static void main(String arg[])
    {
        MyClass MyObj=new MyClass();
        MyObj.myFun(3, 4);
    }
}

```

(3) 将文件保存起来，命名为 MyClass.java，保存在自己工作的目录下。

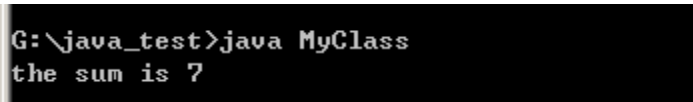
(4) 进入命令方式 (MS—DOS)，并转. java 文件所在目录。敲入下述命令，编译上述 Java 文件。

命令格式: javac MyClass.java

(5) 利用 Java 解释器运行这个 Java Application 程序并查看运行结果。

命令格式: java MyClass

以上程序运行结果如图 1-1 所示。



```

G:\java_test>java MyClass
the sum is 7

```

图 1-1

指导内容 2:

编写并编译第一个 Java Applet 程序。

(1) 打开一个纯文本编辑器 (如记事本)，键入如下程序 (注意大小写):

```

import java.awt.Graphics;
import java.applet.Applet;
public class MyFirstJavaApplet extends Applet
{
    public String s;
    public void init()
    {
        s=new String("Hello World !");
    }
    public void paint(Graphics g)
    {
        g.drawString(s, 25, 25);
    }
}

```

(2) 把文件保存起来，命名为 MyFirstJavaApplet.java，保存在自己测试的目录下。

(3) 进入命令方式 (MS—DOS) 并转. java 文件所在目录，敲入下述命令，编译上述 Java 文件。

命令格式: javac MyFirstJavaApplet.java

(4) 编写配合 Applet 的 HTML 文件, 代码如下:

```
<html>
<head>
<title>My Applet</title>
</head>
<body>
<applet code="MyFirstJavaApplet.class" width=400 height=100>
</applet>
</body>
</html>
```

(5) 将上述内容存盘为 MyFirstJavaApplet.html, 与文件 MyFirstJavaApplet.java 保存在本实验的工作目录下。

(6) 用模拟的 Applet 运行环境解释运行这个 Java Applet 程序并观察运行结果。

命令格式: AppletViewer MyFirstJavaApplet.html

以上程序的运行结果如图 1-2 所示。

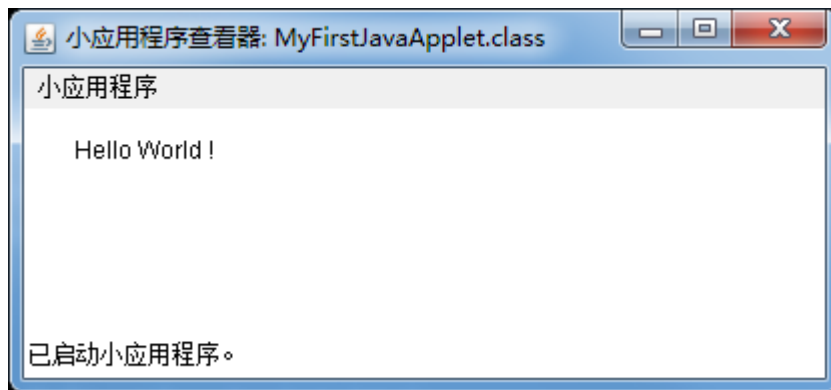


图 1-2

实验二 面向对象的编程技术

一、实验目的

- (1) 掌握类与对象的基本概念以及它们之间的关系;
- (2) 掌握定义类与创建对象实例的方法;

- (3) 掌握类方法和属性的定义和使用；
- (4) 掌握构造方法的定义及其使用。

二、实验内容

1. 基本指导

指导内容：

定义一个类-圆，并编一个主类测试它。

实验步骤：

- (1) 开机后，在 java 实验目录下创建 test4 子目录。本阶段的 Java 源程序、编译后的字节码文件都放在这个目录中。
- (2) 打开一个纯文本编辑器，定义一个类-圆，代码如下：

```
class CCircle {  
  
    double pi;  
  
    double radius;  
  
    double getRadius(){  
  
        return radius;  
  
    }  
  
    void setCircle(double r, double p){  
  
        pi=p;  
  
        radius=r;  
  
    }  
  
}
```

(3) 在上面的代码后面添加主类代码，创建类-圆的一个实例，并输出该圆的半径：

```
public class TestCCircle{

    public static void main(String args[])    {

        CCircle cir1=new CCircle();

        cir1.setCircle(2.0,3.1416);

        System.out.println("radius="+cir1.getRadius());

    }

}
```

(4) 把文件命名为 TestCCircle.java，保存在 java 实验目录的 test4 子目录下。

(5) 编译并运行该程序，程序的运行结果如图 4.1 所示

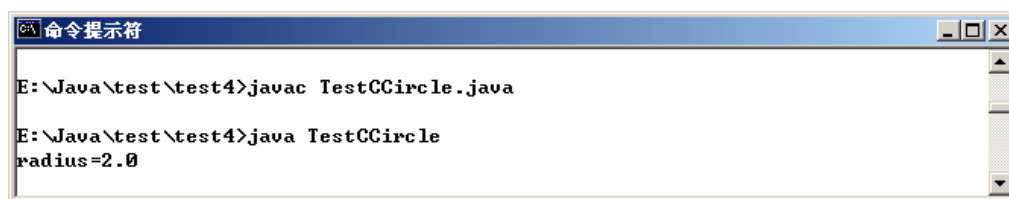


图 4.1

2. 练习思考

练习内容：

扩展圆的定义，为其增加可以求圆面积的方法，并在主类中输出一个实例化的圆的面积。

思考问题：

运行扩展后的程序，思考如下的问题：

- (1) 是否可以将类-圆的定义和主类的源代码放在两个文件中。如果可以的话，两个文件的命名有何要求，上机测试后，给出结论。
- (2) 修改程序，使圆的属性 pi 定义为最终变量，其值为 3.14159，看会出现什么样的结果。如果程序出错，请调整代码以适合属性 pi 为最终变量的要求。
- (3) 为程序添加构造方法代码，调用该构造方法，可以完成圆的半径的初始化。
- (4) 修改主类代码，测试构造方法的使用。

3. 上机作业

(1) 编写 Book.java, 定义一个类 Book, 具有以下属性和方法:

属性: 书名(Title); 出版日期(Pdate); 字数(Words)。

方法: 计算单价 price(): 单价=字数/1000*35*日期系数

上半年的日期系数=1.2; 下半年的日期系数=1.18

(2) 编写主类 TestClass, 在主类中实例化类 Book, 并输出实例化对象的属性。运行该程序, 分析运行的结果, 你觉得你学到了什么?

实验三 类的方法

一、实验目的

- (1) 掌握一维数组的声明、初始化和引用;
- (2) 熟练使用 if/else 语句和 switch 条件分支语句编程;
- (3) 熟练使用 while 语句、do-while 语句、for 语句等循环语句编程。

二、实验内容

1. 基本指导

指导内容 1:

比较两个数的大小并按升序输出。

实验步骤:

(1) 开机后, 在 java 实验目录下创建 test3 子目录。本阶段的 Java 源程序、编译后的字节码文件都放在这个目录中。

(2) 打开一个纯文本编辑器, 键入如下程序 (注意大小写):

```
public class Sort {  
  
    public static void main (String args[]) {  
  
        double d1=23.4;
```

```
double d2=35.1;

if (d2>=d1)

    System.out.println(d2+">="+d1);

else

    System.out.println(d1+">="+d2);

}

}
```

(3) 将文件保存起来，命名为 Sort.java，保存在 java 实验目录的 test3 子目录下。

(4) 进入命令方式 (MS—DOS)，并转.java 文件所在目录。敲入下述命令，编译上述 Java 文件。

命令格式：javac Sort.java

(5) 利用 Java 解释器运行这个 Java Application 程序并查看运行结果。程序的运行结果如图 3.1 所示。

命令格式：java Sort



图 3.1

指导内容 2:

编写程序，输出 1 到 1000 之间，所有可以被 3 整除又可以被 7 整除的数。

(1) 打开一个文本编辑器，键入如下程序（注意大小写）：


```
public class NumTest{

    public static void main (String args[]) {

        int n,num,num1;

        System.out.println("在 1~1000 可被 3 与 7 整除的为");

        for (n=1;n<=1000;n++) {

            num =n%3;

            num1=n%7;

            if (num==0) {

                if (num1==0)

                    System.out.print(n+" ");

            }

        }

        System.out.println(" ");

    }

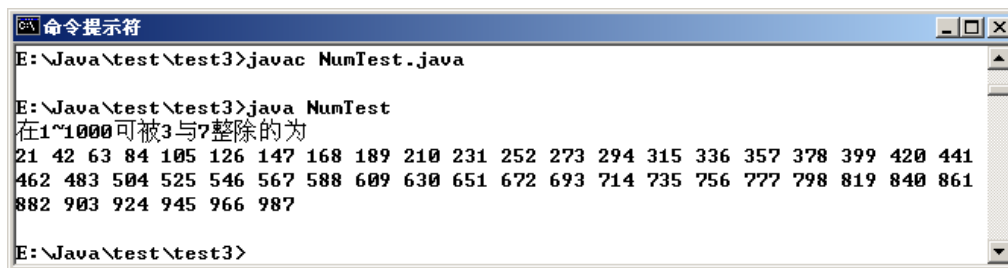
}
```

- (2) 把文件命名为 NumTest.java，保存在 java 实验目录的 test3 子目录下。
- (3) 进入命令方式 (MS—DOS) 并转.java 文件所在目录，敲入下述命令，编译上述 Java 文件。

命令格式：javac NumTest.java

- (4) 利用 Java 解释器运行这个 Java Application 程序并查看运行结果。程序的运行结果如图 3.2 所示。

命令格式：java NumTest。



```
命令提示符
E:\Java\test\test3>javac NumTest.java

E:\Java\test\test3>java NumTest
在1~1000可被3与7整除的为
21 42 63 84 105 126 147 168 189 210 231 252 273 294 315 336 357 378 399 420 441
462 483 504 525 546 567 588 609 630 651 672 693 714 735 756 777 798 819 840 861
882 903 924 945 966 987

E:\Java\test\test3>
```

图 3.2

2. 练习思考

练习内容：

创建一个具有 5 个值的数组，并找出最大值和最小值。

程序代码：

```
public class ArrSort {

    public static void main(String[] args) {

        int arr[]=new int[5];

        int i;

        arr[0]=10;

        arr[1]=20;

        arr[2]=-9;

        arr[3]=8;

        arr[4]=98;

        int min=0,max=0;

        for(i=0;i<5;i++){

            if(max<arr[i])

                max=arr[i];
```

```
        if(min>arr[i])

            min=arr[i];

    }

    System.out.println("数组的最大值是:" +max);

    System.out.println("数组的最小值是:"+min);

}

}
```

思考问题：

- (1) 将上面的数组进行排序，数组的第一个元素为最小值，最后一个元素为最大值。
- (2) 能根据给定的数组值，找出该数组值在数组中的下标。

3. 上机作业

- (1) 编写一个换算 GPA 的 Application 程序，对于学生学习的每门课程，都输入两个整数：

考试成绩和学分，考试成绩按如下公式换算：

85~100： 4

75~84： 3

60~74： 2

45~59： 1

44 以下： 0

GPA 等于换算后每门课的成绩的学分加权平均值（ $\sum（成绩 \times 学分） / \sum 学分$ ）。

学生信息参考下表

| 姓名 | 课程名 | 学分 | 成绩 |
|----|-----|----|------|
| 张一 | 数学 | 4 | 71.5 |

| | | | |
|----|------|-----|------|
| 张二 | 英语 | 3.5 | 80.4 |
| 张三 | 数据结构 | 3 | 95.5 |
| 李一 | 数学 | 4 | 78.5 |
| 李二 | 数据结构 | 3 | 54.5 |
| 李三 | 数据库 | 3 | 60.5 |
| 赵一 | 数学 | 4 | 88.5 |
| 赵二 | 英语 | 3.5 | 92.5 |
| 赵三 | 数据结构 | 3 | 71.5 |

实验四 类的重用

一、实验目的

- (1) 掌握类的继承、方法的继承和重写；
- (2) 掌握 JAVA 基础类库（String 类）的使用；

二、实验内容

1. 基本指导

指导内容 1:

编写一个 Java 应用程序，判断两个字符是否相同，判断字符串的前缀，后缀是否和某个字符串相同，按字典顺序比较两个字符串的大小关系，检索字符串，创建子字符串，将数字型字符串转换为数字，将字符串放到数组中，用字符数组创建字符串。

参考程序模板:

```
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 valueOf(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]);  
        }  
    }  
}
```

3. 上机作业

1. 编写一个 Java 应用程序，除了主类外，该程序中还有 4 个类：People、ChinaPeople、AmericanPeople 和 BeijingPeople 类。要求如下：

- People 类有访问权限是 protected 的 double 型成员变量:height 和 weight，以及 public void speakHello()、public void averageHeight() 和 public void averageWeight()方法
- ChinaPeople 类是 People 的子类，新增了 public void chinaGongfu()方法。要求 ChinaPeople 重写父类的 public void speakHello()、public void averageHeight()和 public void averageWeight()方法
- AmericanPeople 类是 People 的子类，新增 public void americanBoxing()方法，要求 AmericanPeople 重写父类的 public void speakHello()、public void averageHeight()和 public void averageWeight()方法
- BeijingPeople 类是 ChinaPeople 的子类，新增 public void beijingOpera()方法，要求 BeijingPeople 重写父类的 public void speakHello()、public void averageHeight()和 public void averageWeight()方法

参考程序模板：

```
class People  
{ protected double weight,height;  
public void speakHello()  
{System.out.println("yayawawa");  
}  
public void averageHeight()  
{height=173;  
System.out.println("average height:"+height);  
}  
public void averageWeight()  
{
```

```

weight = 70;
System.out.println("average weight:" + weight);
}
}

class ChinaPeople extends People
{
    [代码 1]//重写 public void speakHello()方法, 要求输出"你好, 吃饭了吗"汉语信息
    [代码 2]//重写 public void averageHeight()方法, 要求输出"中国人的平均身高: 173.0 厘米"汉语信息
    [代码 3]//重写 public void averageWeight()方法, 要求输出"中国人的平均体重: 67.34 公斤"汉语信息
    public void chinaGongfu()
    {
        [代码 4]//输出中国武术信息, 例如: "坐如钟, 站如松, 睡如弓"
    }
}

class AmericanPeople extends People
{
    [代码 5]//重写 public void speakHello()方法, 要求输出"How do you do"
    [代码 6]//重写 public void averageHeight()方法
    [代码 7]//重写 public void averageWeight()方法
    public void chinaGongfu()
    {
        [代码 8]//输出拳术信息, 例如: "直拳、勾拳"
    }
}

class BeijingPeople extends ChinaPeople
{
    [代码 9]//重写 public void speakHello()方法, 要求输出"您好, 这里是北京"
    [代码 10]//重写 public void averageHeight()方法
    [代码 11]//重写 public void averageWeight()方法
    public void chinaGongfu()
    {
        [代码 12]//输出京剧信息, 例如: "京剧术语"
    }
}

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

```

```
ChinaPeople chinaPeople = new ChinaPeople();
AmericanPeople americanPeople = new AmericanPeople();
BeijingPeople beijingPeople = new BeijingPeople();
chinaPeople.speakHello();
    americanPeople.speakHello();
    beijingPeople.speakHello();
chinaPeople.averageHeight();
americanPeople.averageHeight();
beijingPeople.averageHeight();
chinaPeople.averageWeight();
americanPeople.averageWeight();
beijingPeople.averageWeight();
    chinaPeople.chinaGongfu();
    americanPeople.americanBoxing();
    beijingPeople.beijingOpera();
    beijingPeople.chinaGongfu();
}
}
```

实验五 包、接口、类库

一、实验目的

- (1) 掌握创建包与引用包的方法；
- (2) 掌握用接口实现多重继承的机制；

二、实验内容

1. 基本指导

指导内容 1:

包的创建和引用。

实验步骤:

- (1) 开机后，在 java 实验目录下创建 test5 子目录。本阶段的 Java 源程序都放在这个子目录中。字节码文件则根据建包的情况放在 test5 相应的子目录中。

(2) 打开一个纯文本编辑器，输入如下的代码：

```
package p1;

public class DefiPackage {

    public void display(){

        System.out.println("in method display()");

    }

}
```

(3) 将文件命名为 DefiPackage.java，保存在 java 实验目录的 test5 子目录下。

(4) 打开 MS-DOS 窗口，转到 DefiPackage.java 所在的目录，键入命令：

```
javac -d . DefiPackage.java
```

(5) 键入 Dir 命令，可以看到在 test5 子目录下创建了 p1 的子文件夹。接着键入下面的命令以查看 p1 下的文件，可以看到 DefiPackage.class 存储在此文件夹下。

```
cd p1
```

```
dir
```

(3)、(4)、(5) 的操作步骤如图 5-1 所示。

```
命令提示符

E:\Java\test\test5>javac -d . DefiPackage.java

E:\Java\test\test5>dir
驱动器 E 中的卷没有标签。
卷的序列号是 78DE-5253

E:\Java\test\test5 的目录

2004-10-30 05:32p    <DIR>      .
2004-10-30 05:32p    <DIR>      ..
2004-10-30 09:53p             121 DefiPackage.java
2004-10-30 09:59p    <DIR>      p1
                1 个文件      121 字节
                3 个目录  9,972,129,792 可用字节

E:\Java\test\test5>cd p1

E:\Java\test\test5\p1>dir
驱动器 E 中的卷没有标签。
卷的序列号是 78DE-5253

E:\Java\test\test5\p1 的目录

2004-10-30 09:59p    <DIR>      .
2004-10-30 09:59p    <DIR>      ..
2004-10-30 10:10p             416 DefiPackage.class
                1 个文件      416 字节
                2 个目录  9,972,129,792 可用字节
```

图 5.1

(6) 在另一个文件中输入如下的代码：

```
import p1.DefiPackage;

public class TestPackage {

    public static void main(String[] args) {

        DefiPackage t=new DefiPackage();

        t.display();

    }

}
```

(7) 把文件命名为 TestPackage.java，保存在 java 实验目录的 test5 子目录下。

(8) 编译并运行该程序，程序的运行结果如图 5.2 所示



```
命令提示符
E:\Java\test\test5>javac TestPackage.java

E:\Java\test\test5>java TestPackage
in method display<>

E:\Java\test\test5>
```

图 5.2

- (9) 在文件 TestPackage.java 中加入包定义语句: [package p2;], 重新正确地编译和运行该程序, 从中理解包的概念。

2. 练习思考

练习内容:

创建接口 Speakable 和 Runner, 然后创建两个类 Dog 和 Person 实现该接口。

程序代码:

```
interface Speakable{

    public void speak();

}

interface Runner{

    public void run();

}

class Dog implements Speakable,Runner{

    public void speak(){

        System.out.println("狗的声音:汪、汪!");

    }

    public void run(){

        System.out.println("狗用四肢跑步");

    }

}
```

```

    }

}

class Person implements Speakable,Runner{

    public void speak(){

        System.out.println("人们见面时经常说:您好! ");

    }

    public void run(){

        System.out.println("人用两腿跑步");

    }

}

public class TestInterface{

    public static void main(String[] args) {

        Dog d=new Dog();

        d.speak(); d.run();

        Person p=new Person();

        p.speak();  p.run();

    }

}

```

思考问题：

运行上面的程序，思考如下的问题：

- (1) 该程序编译后生成几个字节码文件？
- (2) 创建一个类 Bird（鸟），给出其声音特征，并在主类中创建一个 Bird 类的实例，输出

其特征。

- (3) 如何编写抽象类代替程序中的接口，实现程序同样的功能。试比较它们的不同。

3. 上机作业

- (1) 创建一个名称为 `Vehicle` 的接口，在接口中添加两个带有一个参数的方法 `start()` 和 `stop()`。在两个名称分别为 `Bike` 和 `Bus` 的类中实现 `Vehicle` 接口。创建一个名称为 `interfaceDemo` 的类，在 `interfaceDemo` 的 `main()` 方法中创建 `Bike` 和 `Bus` 对象，并访问 `start()` 和 `stop()` 方法。
- (2) 创建一个名称为 `MainPackage` 的包，使它包含 `ParentClass` 和 `SubClass`。`ParentClass` 包含变量声明，其值从构造函数中输出。`SubClass` 类从父类派生而来，完成对父类变量的赋值。创建一个名称为 `DemoPackage` 的主类，使它不在 `MainPackage` 包中，在该类中创建一个 `SubClass` 类的对象。

实验六 异常处理

一、实验目的

- (1) 掌握异常的概念及异常处理的机制；
- (2) 掌握 `try-catch-finally` 异常处理语句的使用；
- (3) 熟悉用户自定义异常及处理用户自定义异常的方法。

二、实验内容

1. 基本指导

指导内容：

编写一个程序，同时捕获数组越界和被 0 除的异常，说明异常处理语句 `try-catch-finally` 的处

理机制。

实验步骤：

(1) 开机后，在 java 实验目录下创建 test11 子目录。本阶段的 Java 源程序及编译生成的字节码文件都放在这个子目录中。

(2) 新建一个 Java 文件，输入如下的程序代码：

```
public class Catch Demo {

    public static void main(String[] args) {

        int number[]={4,8,16,32,64,128,256,512};

        int denom[]={2,0,4,4,0,8};

        for(int i=0;i<number.length;i++){

            try {

                System.out.println(number[i] + " / " + denom[i] + "is " +

                                    number[i] / denom[i]);

            } catch(ArithmeticException exc){

                System.out.println("Can't be divided be zero");

            }

            catch(ArrayIndexOutOfBoundsException exc){

                System.out.println("No matching element found.");

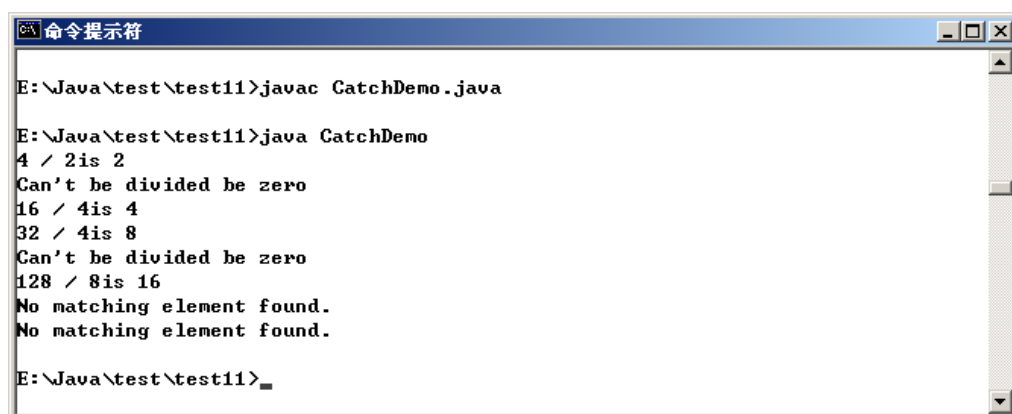
            }

        }

    }

}
```

- (3) 将文件命名为 CatchDemo.java，保存在本次实验目录下并编译并运行该程序，程序的运行结果如图 11-1 所示：



```
E:\Java\test\test11>javac CatchDemo.java

E:\Java\test\test11>java CatchDemo
4 / 2 is 2
Can't be divided by zero
16 / 4 is 4
32 / 4 is 8
Can't be divided by zero
128 / 8 is 16
No matching element found.
No matching element found.

E:\Java\test\test11>
```

图 11-1

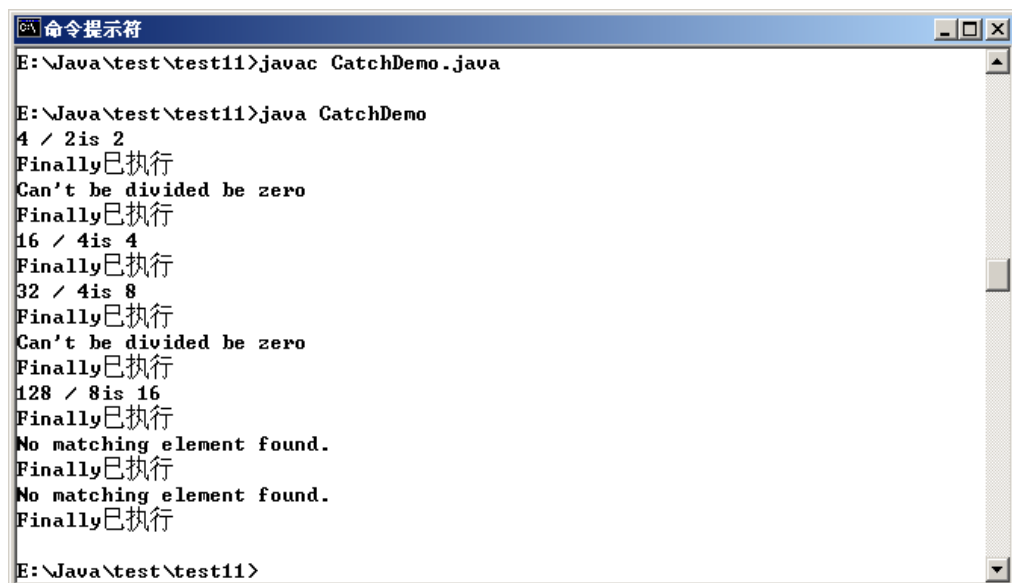
- (4) 为上述的异常处理添加 finally 块，其代码如下：

```
finally{

    System.out.println("Finally 已执行");

}
```

- (5) 重新编译运行该 java 程序，程序的运行结果如图 11-2 所示：
- (6) 试一试：如果没有异常处理，直接输出两个数组对应元素相除的结果，会出现什么样的结果，分析其原因。



```
E:\Java\test\test11>javac CatchDemo.java

E:\Java\test\test11>java CatchDemo
4 / 2 is 2
Finally 已执行
Can't be divided by zero
Finally 已执行
16 / 4 is 4
Finally 已执行
32 / 4 is 8
Finally 已执行
Can't be divided by zero
Finally 已执行
128 / 8 is 16
Finally 已执行
No matching element found.
Finally 已执行
No matching element found.
Finally 已执行

E:\Java\test\test11>
```

图 11-2

2. 练习思考

练习内容：

创建用户自定义异常，用于描述数据取值范围的错误信息。

程序代码：

```
class UserException extends Exception{

    private int idnumber;

    public UserException(String message,int id){

        super(message);

        this.idnumber=id;}

    public int getId(){

        return idnumber;

    }

}

public class TestException {

    public void regist(int num) throws UserException{

        if(num<0){

            throw new UserException("人数为负值，不合理",3);

        }

        System.out.println("登记人数："+num);

    }

    public void manager(){
```



```
try{

    regist(-100);

}catch(UserException e){

    System.out.println("登记出错，类别：" + e.getId());

}

System.out.println("本次登记操作结束");

}

public static void main(String[] args) {

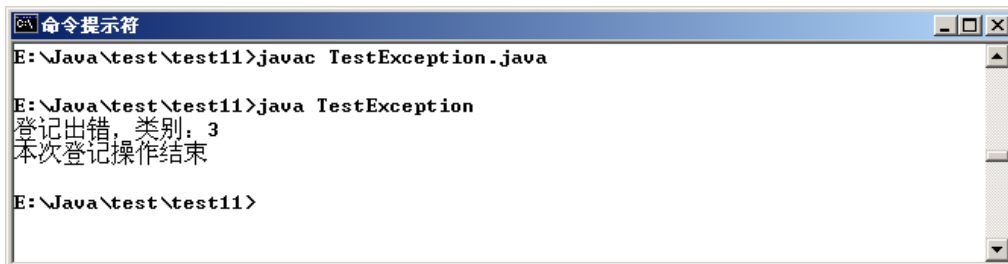
    TestException t=new TestException();

    t.manager();

}

}
```

运行上面的程序，程序的运行结果如图 11-3 所示。



```
命令提示符
E:\Java\test\test11>javac TestException.java

E:\Java\test\test11>java TestException
登记出错, 类别: 3
本次登记操作结束

E:\Java\test\test11>
```

图 11-3

思考问题:

- (1) 本程序中 throws 和 throw 语句的作用是什么?
- (2) 本程序中是如何定义用户自定义异常的?

(3) 本程序是如何处理程序产生的用户自定义异常的?

(4) 如果将程序中的 “`public void regist(int num) throws MyException`” 改为 “`public void regist(int num)` ”, 会出现什么样的情况?

3. 上机作业

编写一个程序, 将字符串转换成数字。请使用 `try-catch-finally` 语句处理转换过程中可能出现的异常。

2015 年 3 月