《Java 程序设计》实验报告 11

学生姓名：李季鸿　　班级：2021 级计本（3）班　　学号：20213002624

实验地点：9 教 404　　指导教师：张春元

实验日期：2023-05-15　　共 2 学时

实验环境：Win10+JDK1.8+ IntelliJ IDEA 2022.1.1

1. 实验目的

掌握 I/O 编程一般方法和思想。

2. 实验内容

（1）用集成化开发工具完成实验教材 P99 实验 2 内容。

（2）用集成化开发工具完成实验教材 P102 实验 3 内容。

（3）请用循环语句编写一份九九乘法口诀表，要求输出规范齐整。

3. 实验过程

报告撰写具体要求：截屏显示或直接写出实验 1 至实验 3 的源码和运行结果。

实验内容（1）：

OutputWordMess.java:

```
package _11_.shiyan1;

import java.util.Vector;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: OutputWordMess
 * \* @author: li-jihong
 * \* Date: 2023-05-15 17:11
 */
public class OutputWordMess{
    public static void main(String args[]) {
        Vector<String> allWord,noSameWord;
        WordStatistic statistic =new WordStatistic();
        statistic.setFileName("src/hello.txt");//本地绝对路径

        statistic.wordStatistic(); //statistic 调用 wordStatistic()方法
        allWord=statistic.getAllWord();
        noSameWord=statistic.getNoSameWord();
        System.out.println("共有"+allWord.size()+"个英文单词");
        System.out.println("有"+noSameWord.size()+"个互不相同英文单词");
        System.out.println("按出现频率排列:");
```

```
            int count[]=new int[noSameWord.size()];//判断 noSameWord.elementAt(i)
相同出现的频率
            for(int i=0;i<noSameWord.size();i++) {
                String s1 = noSameWord.elementAt(i);
                for(int j=0;j<allWord.size();j++) {
                    String s2=allWord.elementAt(j);
                    if(s1.equals(s2))
                        count[i]++;
                }
            }
            for(int m=0;m<noSameWord.size();m++) {              //按照频率排序
                for(int n=m+1;n<noSameWord.size();n++) {
                    if(count[n]>count[m]) {
                        String temp=noSameWord.elementAt(m);
                        noSameWord.setElementAt(noSameWord.elementAt(n),m);
                        noSameWord.setElementAt(temp,n);
                        int t=count[m];
                        count[m]=count[n];
                        count[n]=t;
                    }
                }
            }
            for(int m=0;m<noSameWord.size();m++) {
                double frequency=(1.0*count[m])/allWord.size();
                System.out.printf("%s:%-7.3f",noSameWord.elementAt(m),frequency);
            }
        }
}
```

WordStatistic.java:

```
package _11_.shiyan1;

import java.io.File;
import java.io.IOException;
import java.util.Scanner;
import java.util.Vector;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: WordStatistic
 * \* @author: li-jihong
 * \* Date: 2023-05-15 17:11
 */
```

```
public class WordStatistic {
    Vector<String> allWord, noSameWord;
    File file = new File("src/english.txt");//按照实际路径创建该文本
    Scanner sc = null;
    String regex;

    WordStatistic() {
        allWord = new Vector<String>();
        noSameWord = new Vector<String>();
        //regex 是由空格、数字和符号(!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~)组成的
正则表达式
        regex = "[\\s\\d\\p{Punct}]+"; //   p187 正则表达式
        try {
            sc = new Scanner(file); //创建指向 file 的 sc
            sc.useDelimiter(regex); //sc 调用 useDelimiter(String regex)方法,向参
数传递 regex
        } catch (IOException exp) {
            System.out.println(exp.toString());
        }
    }

    void setFileName(String name) {
        file = new File(name);
        try {
            sc = new Scanner(file);
            sc.useDelimiter(regex);
        } catch (IOException exp) {
            System.out.println(exp.toString());
        }
    }

    public void wordStatistic() {
        try {
            while (sc.hasNext()) {
                String word = sc.next();
                allWord.add(word);
                if (!noSameWord.contains(word)) noSameWord.add(word);
            }

        } catch (Exception e) {
        }
    }
```

```
    public Vector<String> getAllWord() {
        return allWord;
    }

    public Vector<String> getNoSameWord() {
        return noSameWord;
    }
}
```

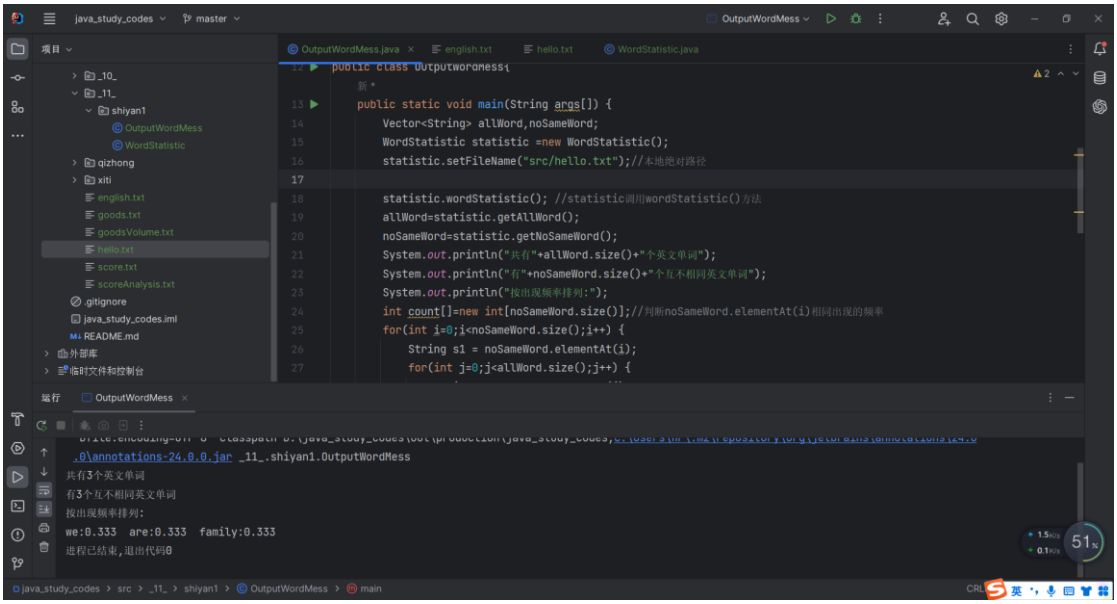


图 11-1 实验一

实验后的练习

按字典顺序输出全部不相同的单词。

程序源代码：

```
    import java.util.*;
    public class DictionaryOutput {
        public static void main(String args[]) {
            Vector<String> allWord,noSameWord;
            WordStatistic statistic=new WordStatistic();
            statistic.setFileName("src/hello.txt");
            statistic.wordStatistic();
            allWord=statistic.getAllWord();
            noSameWord=statistic.getNoSameWord();
            System.out.println("一共有"+allWord.size()+"个英文单词");
            System.out.println("有"+noSameWord.size()+"个不相同的英文单词
");
```

```
            System.out.println("按字典顺序排列:");
            String s[]=new String [noSameWord.size()];
            for(int i=0; i<noSameWord.size();i++) {
                s[i]=noSameWord.elementAt(i);
            }
            Arrays.sort(s);
            for(int i=0;i<noSameWord.size();i++) {
                System.out.println(s[i]+" ");
            }
        }
    }
```

程序运行结果：

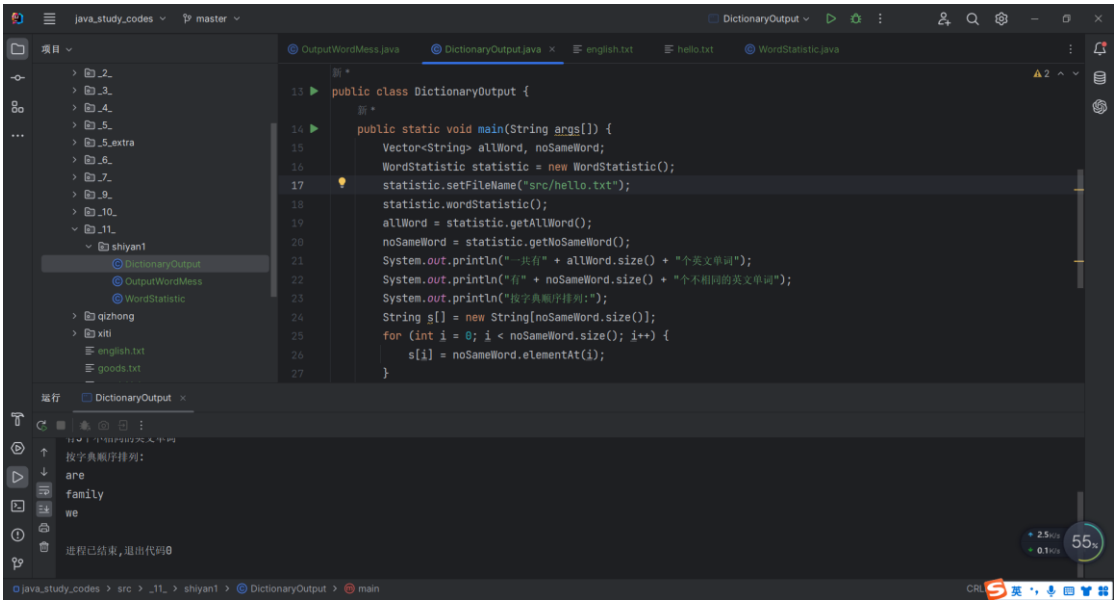


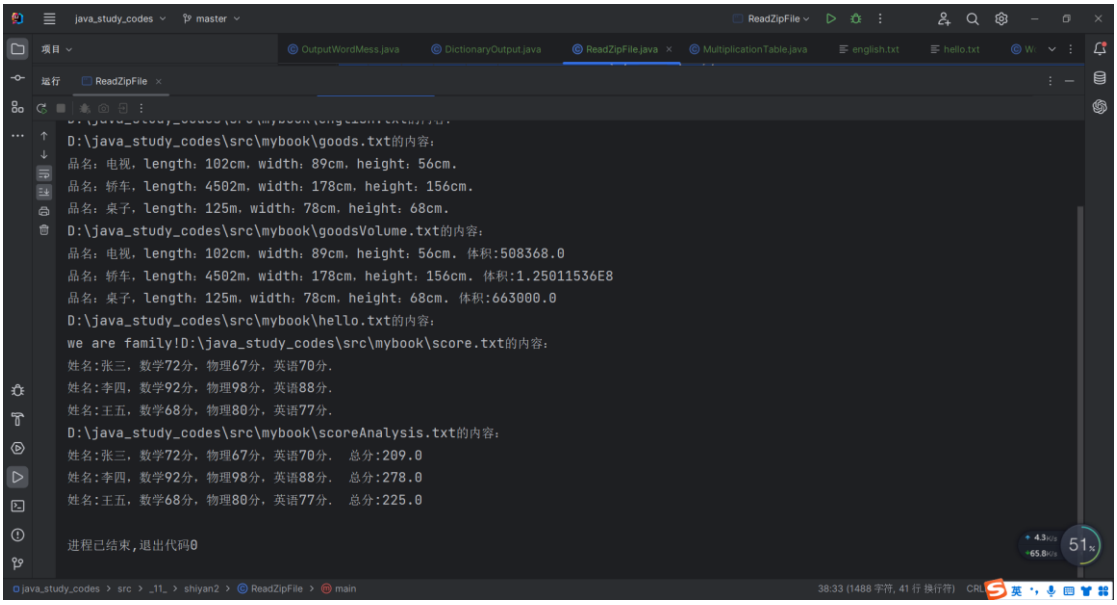
图 11-2　实验 1 课后练习

实验内容（2）：

```
package _11_.shiyan2;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.zip.ZipEntry;
import java.util.zip.ZipInputStream;
```

```
/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: ReadZipFile
 * \* @author: li-jihong
 * \* Date: 2023-05-15 17:25
 */
public class ReadZipFile {
    public static void main(String args[]) {
        File f = new File("src/book.zip");
        File dir = new File("src/mybook");
        byte b[] = new byte[1024];
        dir.mkdir();
        try {
            ZipInputStream in = new ZipInputStream(new FileInputStream(f));
            ZipEntry zipEntry = null;
            while ((zipEntry = in.getNextEntry()) != null) {
                File file = new File(dir, zipEntry.getName());
                FileOutputStream out = new FileOutputStream(file);
                int n = -1;
                System.out.println(file.getAbsolutePath() + "的内容：");
                while ((n = in.read(b)) != -1) {
                    out.write(b, 0, n);
                    System.out.print(new String(b, 0, n, "UTF-8"));
                }
                out.close();
            }
            in.close();
        } catch (IOException ee) {
            System.out.println(ee);
        }
    }
}
```



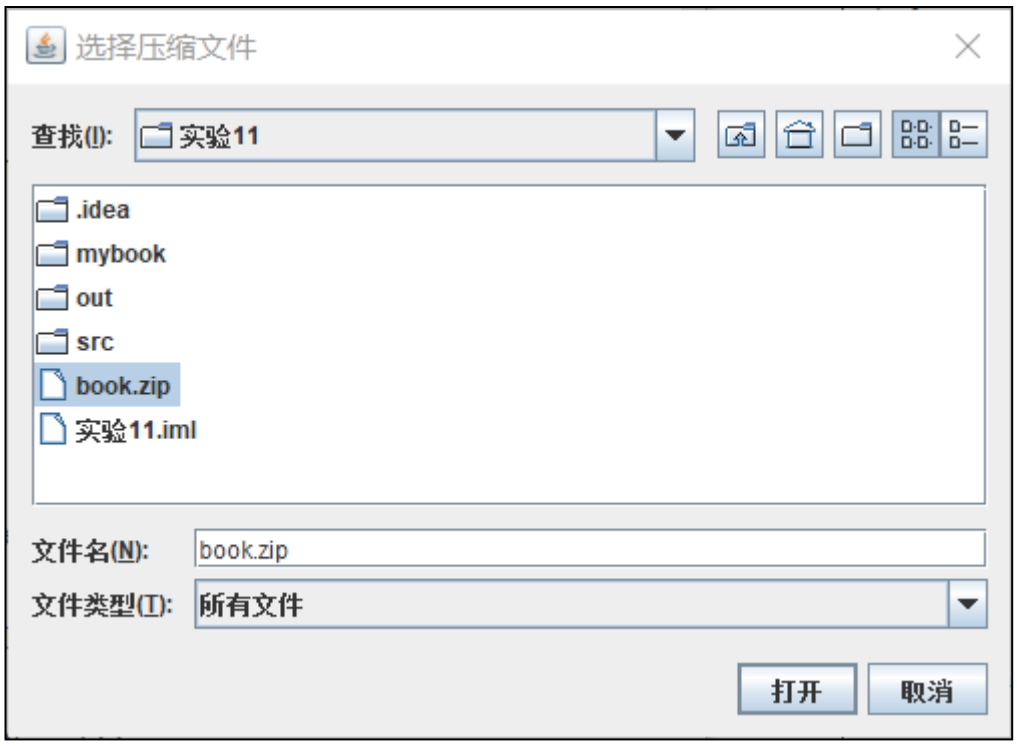
实验后的练习：

```
public class ReadZipFile {
    public static void main(String args[]){
        byte b[] = new byte[100];
        JFileChooser jfc = new JFileChooser();
        jfc.setDialogTitle("选择压缩文件");
        jfc.setDialogType(JFileChooser.OPEN_DIALOG);
        int r = jfc.showOpenDialog(new JFileChooser());
        if(r==JFileChooser.APPROVE_OPTION){
            File file = jfc.getSelectedFile();
            try{
                ZipEntry zipEntry = null;
                ZipInputStream in = new ZipInputStream(new FileInputStream(file), Charset.forName("GBK"));

                JFileChooser jfc_rec = new JFileChooser();
                jfc_rec.setDialogTitle("选择存放的位置");
                jfc_rec.setDialogType(JFileChooser.OPEN_DIALOG);
                int r1 = jfc_rec.showSaveDialog(new JFileChooser());
                if(r1 == JFileChooser.APPROVE_OPTION){
                    File dir = new File(jfc_rec.getSelectedFile().getAbsolutePath());
                    dir.mkdir();
                    JOptionPane.showMessageDialog(null, "文件将存放在"+dir.getAbsolutePath()+"文件夹中，是否确认", "确认存放地址", JOptionPane.WARNING_MESSAGE);
                    while((zipEntry = in.getNextEntry())!=null){
                        File parentFile = new File(dir,zipEntry.getName());
                        FileOutputStream out = new FileOutputStream(parentFile);
```

```
                        int n = -1;
                        System.out.println(file.getAbsolutePath()+"的内容：");
                        while((n=in.read(b,0,100))!=-1){
                            String str = new String(b,0,n);
                            System.out.println(str);
                            out.write(b,0,n);
                        }
                        out.close();
                    }
                    in.close();
                }
            }
            catch (Exception exp){

            }
        }
    }
}
```



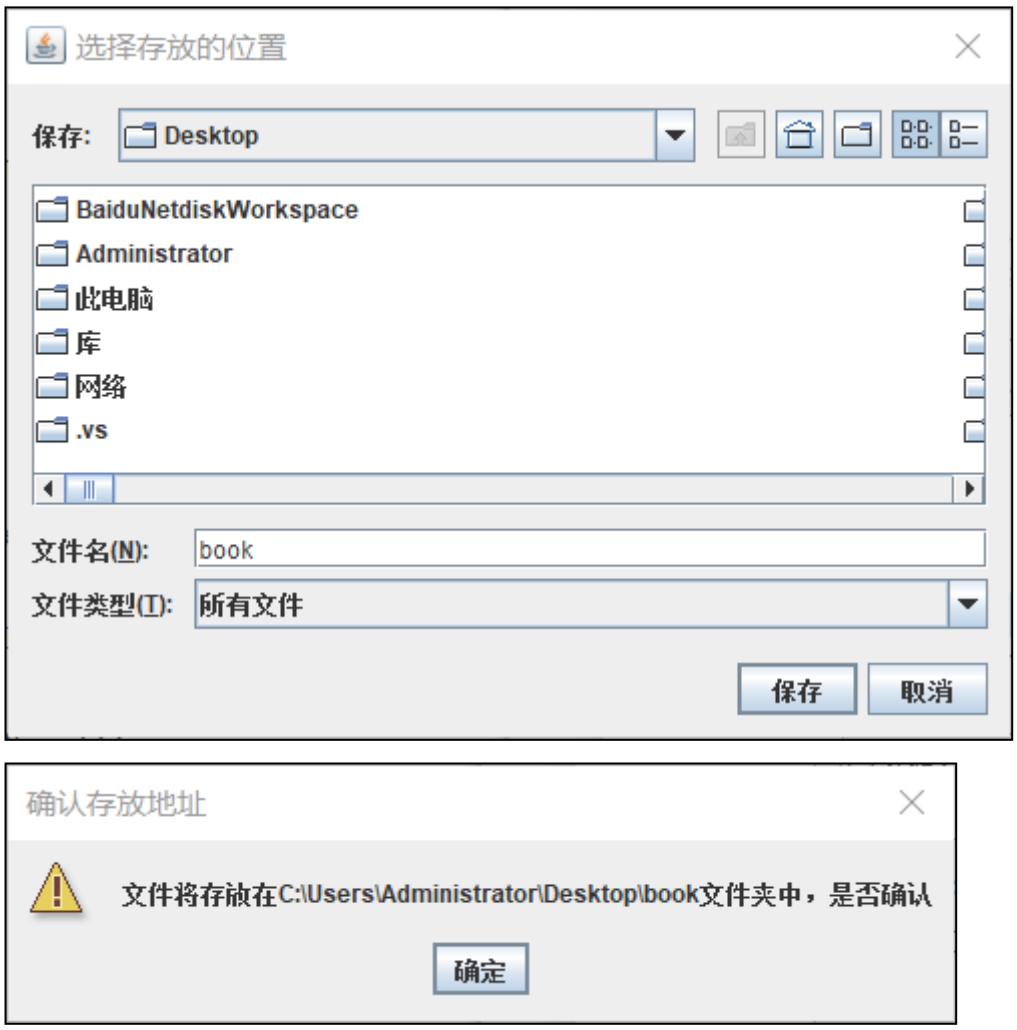


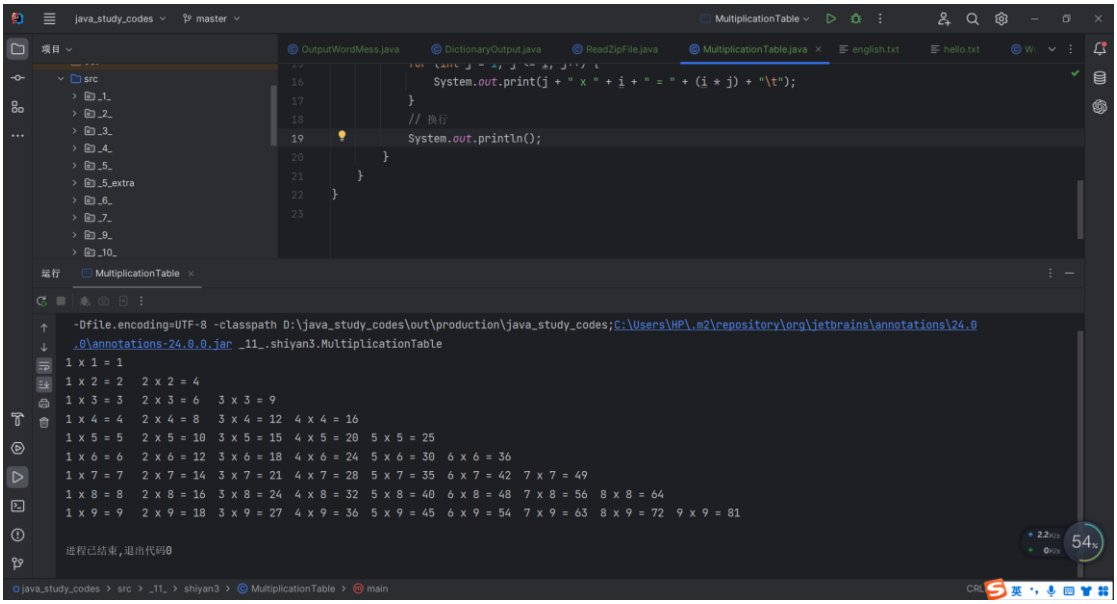
图 1-2 实验结果

实验内容（3）：

```
package _11_.shiyan3;

/**
 * \* Created with IntelliJ IDEA.
 * \* @ProjectName: java_study_codes
 * \* @FileName: MultiplicationTable
 * \* @author: li-jihong
 * \* Date: 2023-05-15 17:38
 */
public class MultiplicationTable {
    public static void main(String[] args) {
        // 控制乘数
        for (int i = 1; i <= 9; i++) {
            // 控制被乘数
```

```
                for (int j = 1; j <= i; j++) {
                    System.out.print(j + " x " + i + " = " + (i * j) + "\t");
                }
                // 换行
                System.out.println();
            }
        }
}
```



4. 实验总结

写出实验中的心得体会（对第 10 章理论课重点简述）。