

实验报告

实验名称	实验 5 字符串与输入输出			实验时间	第 12 周		
实验环境							
Eclipse/Jcreator Pro/JDK。							
实验目的和内容要求							
<p>一、实验目的</p> <ol style="list-style-type: none">1. 掌握字符串常用函数的使用方法。2. 了解输入输出的基本原理；掌握类 File、FileInputStream、FileOutputStream、RandomAccessFile 的使用方法。 <p>二、实验要求</p> <ol style="list-style-type: none">1. 提供所完成的各道题主要实验代码和运行结果的界面截图。2. 简述在实验过程中遇到的问题与解决方法。3. 简述实验过程中的发现与收获，未解决或需进一步解决的问题。 <p>三、实验内容</p> <ol style="list-style-type: none">1. 输入一个字符串，将其中的大写英文字母改为小写字母，小写字母改为大写字母，然后输出。2. 检查输入的字符串是否是“回文”，例如“abc123321cba”就是回文。3. 输入两个字符串，从第一个字符串中删除第二个字符串的内容。4. 从命令行方式输入的字符串中删去所有重复的字符，每种字符只保留一个。5. 统计一个字符串中给定字符串出现的频率。6. 将一个表示十进制数的字符串转换为以逗号分隔的字符串，从右边开始每三个数字标一个逗号。例如，给定一个字符串“1234567”，该方法返回“1,234,567”。7. 求 2~200 之间的所有素数，将求得的结果保存到 PRIME.DAT 文件中。8. 检查 C 盘根目录下 CONFIG.9. SYS 文件是否存在，若在则显示该文件的名称和内容。10. 输入 5 个学生的信息（包含学号、姓名、3 科成绩），统计各学生的总分，然后将学生信息和统计结果存入二进制数据文件 STUDENT.DAT 中。11. 从第（10）题中建立的 STUDENT.DAT 文件中读取数据，寻找平均分最高的学生，并输出该学生的所有信息。							
实验过程、结果分析与总结							
<p>第 1-9 题</p> <p>源代码：</p>							

```
import java.io.*;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) throws Exception{

        t1();
        t2();
        t2_2();
        t3();
        t4();
        t5();
        t6();
        t7();
        t8t9();

    }

    //将其中的大写英文字母改为小写字母，小写字母改为大写字母，然后输出。
    public static void t1() {

        System.out.println("-----1-----");
        System.out.println("请输入一个字符串：");
        Scanner sc = new Scanner(System.in);
        String str = sc.next();

        String result = "";
        for (int i = 0; i < str.length(); i++) {
            char ch = str.charAt(i);
            if (ch <= 'Z' && ch >= 'A') {
                ch += 32;
            } else if (ch <= 'z' && ch >= 'a') {
                ch -= 32;
            }
            result = result + ch;
        }
        System.out.println("结果是" + result);
    }

    //检查输入的字符串是否是“回文”，
    public static void t2() {

        System.out.println("-----2-----");
        System.out.println("请输入一个字符串：");
        Scanner sc = new Scanner(System.in);
        String str = sc.next();
```

```

int flag = 0;
for (int i = 0; i < (str.length())/2; i++) {
    if (str.charAt(i) != str.charAt(str.length() - 1 - i)) {
        flag = 1;
        break;
    }
}
if (flag == 0) {
    System.out.println(str + "是回文");
} else {
    System.out.println(str + "不是回文");
}
}

```

//第二题再用 StringBuilder 和 链式编程 再写一遍!

```

public static void t2_2() {

    System.out.println("-----2-----");
    System.out.println("请输入一个字符串: ");
    Scanner sc = new Scanner(System.in);
    String str = sc.next();

    //    StringBuilder sb = new StringBuilder();
    //    sb.append(str);
    //    sb.reverse();
    //    String result = sb.toString();

    String result = new StringBuilder(str).reverse().toString();

    if (str.equals(result)) {
        System.out.println(str + "是回文");
    } else {
        System.out.println(str + "不是回文");
    }
}

```

//输入两个字符串，从第一个字符串中删除第二个字符串的内容。

```

public static void t3() {

    System.out.println("-----3-----");
    System.out.println("请输入第一个字符串: ");
    Scanner sc1 = new Scanner(System.in);
    String str1 = sc1.next();
    System.out.println("请输入第二个字符串: ");
    Scanner sc2 = new Scanner(System.in);
    String str2 = sc2.next();
}

```

```

String result = "";
for (int i = 0; i < str1.length(); i++) {
    for (int j = 0; j < str2.length(); j++) {
        if (str1.charAt(i) == str2.charAt(j)) {
            break;
        } else if (j == str2.length() - 1) {
            result = result + str1.charAt(i);
        }
    }
}
System.out.println("结果是" + result);
}

```

//4. 从命令行方式输入的字符串中删去所有重复的字符，每种字符只保留一个。

```

public static void t4() {
    System.out.println("-----4-----");
    System.out.println("请输入一个字符串：");
    Scanner sc = new Scanner(System.in);
    String str = sc.next();

    String result = "";
    result = result + str.charAt(0);
    for (int i = 1; i < str.length(); i++) {
        for (int j = 0; j < result.length(); j++) {
            if (str.charAt(i) == result.charAt(j)) {
                break;
            } else if (j == result.length() - 1) {
                result = result + str.charAt(i);
            }
        }
    }
    System.out.println("结果是" + result);
}

```

//统计一个字符串中给定字符串出现的频率。

```

public static void t5() {
    System.out.println("-----5-----");
    System.out.println("请输入一个字符串：");
    Scanner sc = new Scanner(System.in);
    String str = sc.next();
    System.out.println("请输入给定字符串：");
    Scanner sc2 = new Scanner(System.in);
    String str2 = sc2.next();

    int count = 0;

```

```

int j = 0;
for (int i = 0; i < str.length(); i++) {
    if (str.charAt(i) == str2.charAt(j)) {
        j++;
        if (j == str2.length()) {
            count++;
            j = 0;
        }
    } else {
        j = 0;
    }
}
System.out.println("该字符出现的频率为: " + count);
}

```

//将一个表示十进制数的字符串转换为以逗号分隔的字符串，从右边开始每三个数字标一个逗号。

```

public static void t6() {

    System.out.println("-----6-----");
    System.out.println("请输入一个字符串: ");
    Scanner sc = new Scanner(System.in);
    String str = sc.next();

    String result = "";
    int add = (str.length() - 1) / 3;
    int head = str.length() % 3;
    if (head == 0) {
        head = 3;
    }

    if (add != 0) {
        for (int j = 0; j < head; j++) {
            result = result + str.charAt(j);
        }
        result = result + ",";

        for (int j = 0; j < add; j++) {
            for (int k = 0; k < 3; k++) {
                result = result + str.charAt(head + k + 3 * j);
            }
            if (j != add - 1) {
                result = result + ",";
            }
        }
    } else {
        result = str;
    }
}

```

```

    }

    System.out.println("结果是" + result);
}

//7. 求 2~200 之间的所有素数，将求得的结果保存到 PRIME.DAT 文件中。
public static void t7() throws IOException {
    System.out.println("-----7-----");
    FileOutputStream prime = new FileOutputStream("D:\\PRIME.DAT");

    for (int i = 2; i <= 200; i++) {
        for (int j = 2; j <= i; j++) {
            if (j == i-1 || i == 2) {
//                System.out.println(i);

                //惯例：
                //-----
                String msg = i + " ";
                prime.write(msg.getBytes());
                //-----
                break;
            }
            if (i % j == 0) {
                break;
            }
        }
    }

    prime.close(); //在这才能关闭，而不是在循环中写一个关一下。即关闭和 new
同级
}

//8. 检查 C 盘根目录下 CONFIG.
//9. SYS 文件是否存在，若在则显示该文件的名称和内容。
public static void t8t9() throws IOException {
    System.out.println("-----8 & 9-----");
    File config = new File("D:\\PRIME.DAT");

    if (config.exists()) {
        System.out.println("该文件存在。其名称和内容读取如下");
        System.out.println(config.getName());

        //惯例：
        //-----
        FileInputStream fis = new FileInputStream(config);
        byte[] data = new byte[(int) config.length()];
        fis.read(data);
        fis.close();
    }
}

```

```

        String msg = new String(new String(data));
        //-----
        System.out.println(msg);
    } else {
        System.out.println("文件不存在。");
    }
}
}

```

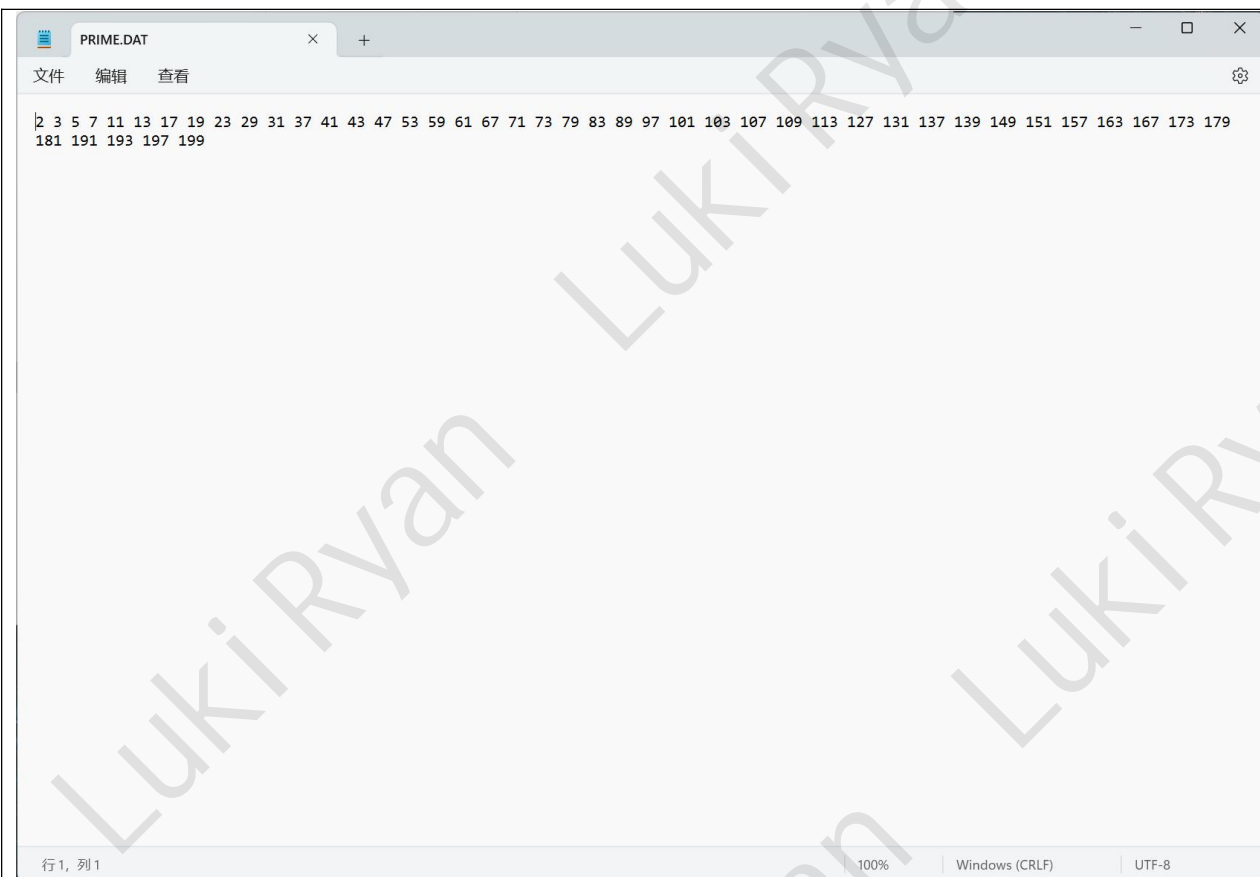
终端截图：

```

运行 Main x
Edition 2023.1.2\bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath C:\Users\33742\...\newLUA\out\production\...
-----1-----
请输入一个字符串:
fewfBRTWcsdfCGREHR
结果是FEWFbrtwCSDfCgrehr
-----2-----
请输入一个字符串:
abc123321cba
abc123321cba是回文
-----2-----
请输入一个字符串:
qweasdqweasdqweasdevbdwjcbwuccbuwuc
qweasdqweasdqweasdevbdwjcbwuccbuwuc不是回文
-----3-----
请输入第一个字符串:
abcabcabcabcdefgdefg
请输入第二个字符串:
bc
结果是aaaadefgdefg
-----4-----
请输入一个字符串:
ZXCZXCZXCZXCZXCZXCasdasdasdasdasdwqequequequeque
结果是zxcasdwqe
-----5-----
请输入一个字符串:
dfguessuicweceyhfdjdsdfjvsjdhfkjdsvhcjewucbsucbjcbscewybycbwucbsdfbbfsdhjvdrvufsec
请输入给定字符串:
df
该字符出现的频率为: 3
-----6-----
请输入一个字符串:
5432342354323435326
结果是5, 432, 342, 354, 323, 435, 326
-----7-----
-----8 & 9-----
该文件存在, 其名称和内容读取如下
PRIME.DAT
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199
进程已结束, 退出代码0

```

PRIME.DAT 文件：



第 10 题

源代码:

Demo10.java

```
import java.io.*;

public class Demo10 {
    public static void main(String[] args) throws Exception {
        Student[] students = {new Student("001", "张三", 89, 90, 91),
            new Student("002", "李四", 92, 93, 94),
            new Student("003", "王五", 95, 96, 97),
            new Student("004", "赵六", 98, 99, 100),
            new Student("005", "田七", 100, 100, 100)};
        ser(students);
    }

    public static void ser(Object obj[]) throws Exception {
        File file = new File("d:\\STUDENT.DAT");
        ObjectOutputStream oos = null;
        OutputStream out = new FileOutputStream(file);
        oos = new ObjectOutputStream(out);
        oos.writeObject(obj);
        oos.close();
    }
}
```



```
}  
}  
  
class Student implements Serializable {  
    private String id;  
    private String name;  
    private int chinese;  
    private int math;  
    private int english;  
  
    public Student() {  
    }  
  
    public Student(String id, String name, int chinese, int math, int english) {  
        this.id = id;  
        this.name = name;  
        this.chinese = chinese;  
        this.math = math;  
        this.english = english;  
    }  
  
    public String getId() {  
        return id;  
    }  
  
    public void setId(String id) {  
        this.id = id;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public int getChinese() {  
        return chinese;  
    }  
  
    public void setChinese(int chinese) {  
        this.chinese = chinese;  
    }  
  
    public int getMath() {  
        return math;  
    }  
}
```

```
{
  english) {
```

g;L nameq ~ xp Y [Zt 001t 0袁执竺sq ~ \ ^]t 002t 0徽床凉sq ~ sq ~ d d dt 005t 0登颂鞍

100% | Unix (LF) | ANSI

第 11 题

Demoll.java

```
import java.io.*;

public class Demoll {
    public static void main(String[] args) throws Exception {
        Object[] obj = dser();
        Student stuMax = (Student) obj[0];
        for (int i = 0; i < obj.length; i++) {
            Student student = (Student) obj[i];
            if
((student.getEnglish()+student.getMath()+student.getEnglish())/3.0>
(stuMax.getChinese()+stuMax.getMath()+stuMax.getEnglish())/3.0) {
                stuMax = student;
            }
        }
        System.out.println("学号: "+stuMax.getId()+" 姓名: "+stuMax.getName() +
" 语文成绩: " + stuMax.getChinese() + " 数学成绩: " + stuMax.getMath() + " 英语
成绩: " + stuMax.getEnglish());
    }

    public static Object[] dser() throws Exception {
        File file = new File("d:\\STUDENT.DAT");
        ObjectInputStream ois = null;
        InputStream input = new FileInputStream(file);
        ois = new ObjectInputStream(input);
        Object[] obj = (Object[]) ois.readObject();
        ois.close();
        return obj;
    }
}
```

运行结果:

C:\Users\3512\桌面\NEWIDEA\001\production\exp0 Demoll

学号: 005 姓名: 田七 语文成绩: 100 数学成绩: 100 英语成绩: 100

进程已结束,退出代码0

通过这次实验，我主要学到了以下内容：

字符串常用函数的使用方法：我学习了如何使用字符串的常用函数，例如 `toUpperCase()`、`toLowerCase()` 等，来实现字符串大小写转换。还学习了如何使用其他字符串函数来完成实验中的其他任务，如判断回文、删除字符串内容等。

输入输出的基本原理和文件类的使用方法：我了解了输入输出的基本原理，包括如何从命令行输入字符串和将结果输出到界面。我还学习了如何使用 `File`、`FileInputStream`、`FileOutputStream` 和 `RandomAccessFile` 等类来操作文件，包括读取文件内容、写入文件和检查文件是否存在等操作。

在实验过程中，我遇到了一些问题，并采取了相应的解决方法：

问题 1：如何使用字符串函数来完成字符串的大小写转换？

解决方法：我学习了如何使用 `toUpperCase()` 和 `toLowerCase()` 函数来实现字符串的大小写转换。通过调用这些函数，我可以将大写字母转换为小写字母，或将小写字母转换为大写字母。

问题 2：如何判断一个字符串是否是回文？

解决方法：我可以使用字符串的 `charAt()` 函数和循环来比较字符串的首尾字符是否相等，从而判断字符串是否是回文。通过设置两个指针，一个指向字符串的开头，一个指向字符串的结尾，并不断比较字符，可以确定字符串是否是回文。

在实验过程中，我有一些发现和收获：

发现：字符串函数提供了丰富的功能，可以方便地操作和处理字符串。掌握字符串函数的使用方法对于处理字符串相关的任务非常重要。

收获：我进一步理解了输入输出的基本原理和文件操作的方法。通过学习 `File`、`FileInputStream`、`FileOutputStream` 和 `RandomAccessFile` 等类的使用，我可以对文件进行读取和写入操作，以及检查文件的存在。

总体而言，通过完成这个实验，我对字符串函数的使用、输入输出的基本原理和文件操作有了初步的了解，并且学到了一些基本的解决问题的方法。我也意识到自己还需要进一步学习和实践，以提高自己在 Java 编程中的能力和熟练度。

指导老师评阅意见

指导老师：

年

月

日

填写内容时，可把表格扩大。