

深圳大学实验报告

课程名称: Java 程序设计

实验项目名称: 必实验 1 基本类型应用

学院: 计算机与软件学院

专业: 计算机科学与技术

指导教师: 姚俊梅

报告人: 刘嘉豪 学号: 2021150102 班级: 高性能班

实验时间: 2023 年 9 月 11 日 (周一)

实验报告提交时间: 2023 年 9 月 11 日 (周一)

教务部制

实验目的与要求：

实验目的：在掌握 JAVA 数组基本概念及应用的基础上，变换数组的内容，完成主类创建，查找等功能的实现；熟练掌握数据类型、运算符、表达式和语句。

实验要求：

(1) 创建一个 100×100 的二维数组，对数组中的元素进行随机赋值（要求使用 `Math.random()` 生成 0-1 之间的浮点数）。通过算法找到该数组中最大的 5 个数，要求从大到小输出，同时计算整个程序所耗费的时间，并分析算法的复杂度（如果复杂度较高会扣 5 分，尽可能用两种方法实现，并进行复杂度对比分析）。在报告中附上程序截图、运行结果截图和详细的文字说明。（40 分）

(2) 某省居民电价分三个“阶梯”：

对于用电量 50 度以内的这部分电量，电价为 0.65 元/度；

对于用电量在 51 度至 220 度之间的这部分电量，电价为 0.68 元/度；

对于用电量超过 220 度的这部分电量，电价为 0.75 元/度。

编写程序，用户从键盘输入用电量（要求在不终止程序的情况下能循环 10 次从键盘读入 10 个数字：48、52、230、90、80、89、510、60、1、10），程序输出用户应缴纳的电费。在报告中附上程序截图、运行结果截图和详细的文字说明。（25 分）

(3) 编写程序：用户从键盘输入 1 个正整数，程序逆序打印出该正整数的各位数字（例如：某个正整数为 123456，程序输出 654321），要求在不终止程序的情况下能循环 10 次从键盘读入。在报告中附上程序截图、运行结果截图和详细的文字说明。（25 分）

报告写作。要求：主要思路有明确的说明，行文逻辑清晰可读性强，报告整体写作较为专业。（10 分）

说明：

(1) 本次实验课作业满分为 100 分，占总成绩的比例（待定）。

(2) 报告正文：请在指定位置填写，本次实验需要单独提交源程序文件（源程序单独打包在 Blackboard 中上传，不要包含外部导入的包，不要和实验报告放到一个压缩包）。

(3) 个人信息：WORD 文件名中的“姓名”、“学号”，请改为你的姓名和学号；实验报告的首页，请准确填写“学院”、“专业”、“报告人”、“学号”、“班级”、“实验报告提交时间”等信息。

(4) 提交方式：实验报告当堂提交，代码运行需找老师或助教进行演示。课后提交按最高 80 分计算。

(5) 发现抄袭（包括复制&粘贴整句话、整张图），抄袭者和被抄袭者的成绩记零分（含抄袭往届同学的作业）。

(6) 延迟提交，酌情扣分；期末考试阶段补交无效。

(1) 创建一个 100×100 的二维数组，对数组中的元素进行随机赋值（要求使用 `Math.random()` 生成 0-1 之间的浮点数）。通过算法找到该数组中最大的 5 个数，要求从大到小输出，同时计算整个程序所耗费的时间，并分析算法的复杂度（如果复杂度较高会扣 5 分）。在报告中附上程序截图、运行结果截图和详细的文字说明。（40 分）

1. 创建二维数组

创建一个 100×100 的 `double` 类型的二维数组。

```
1. //生成 100*100 的二维数组
2. int len = 100;
3. double[][] a = new double[len][len];
```

2. 二维数组初始化

使用 `Math.random()` 生成 0-1 之间的浮点数，对二维数组进行赋值。

```
1. //使用 Math.random() 生成 0-1 之间的浮点数，对矩阵赋值
2. for(int i = 0; i < len; i ++ )
3.     for(int j = 0; j < len; j ++ )
4.         a[i][j] = Math.random();
```

3. 计算整个程序所耗费的时间

我们用 `System.currentTimeMillis()` 函数分别获取程序开始运行时的系统时间和程序结束运行时的系统时间，最后二者相减获取程序的运行时间。

```
1. //计算程序运行时间
2. long ST = System.currentTimeMillis();
3. //中间为核心代码部分
4. //...
5. long ET = System.currentTimeMillis();
6. System.out.print("程序运行时间" + (ET - ST) + "ms");
```

4. 答案的存储工作

接着创建了一个包含五个元素的数组 `ans[]`，并初始化为 -1。这个数组将被用来从大到小地存储最大的五个数。也就是说，数组 `ans[]` 始终是保持非递增地存储最大地五个数。

```
1. //创建数组存储答案，初始化为-1
2. double[] ans = new double[5];
3. for(int i = 0; i < 5; i ++ )
4.     ans[i] = -1;
```

5. 核心算法：找到二维数组中最大的 5 个数

遍历二维数组，对于二维数组中的每个元素，将其从后往前地与数组 `ans[]` 进行比较。首先先与 `ans[]` 最后一个元素 `ans[4]` 比较，如果大于 `ans[4]`，我们就可以把当前二维数组

的元素放入 ans[4]，这个元素新插入的下标我们记为 idx。接着我们需要保持 ans[] 数组的非递增性，也就是说，如果 ans[idx] > ans[idx-1]，我们就需要把二者交换，然后一直从后往前比较，直到比较到 ans[0]。这样 ans[] 中始终是从大到小的存放最大的 5 个数。

```
1. for(int i = 0; i < len; i ++ )
2.     for(int j = 0; j < len; j ++ ) {
3.         // 首先将当前元素 a[i][j] 与 ans[4] 比较
4.         int idx = 4;
5.         if(a[i][j] > ans[idx]) {
6.             ans[idx] = a[i][j];
7.         }
8.         // 如果 ans[idx] > ans[idx - 1], 就交换二者, 同时注意 idx 不能越界
9.         while(idx > 0 && ans[idx] > ans[idx - 1]) {
10.            double t = ans[idx];
11.            ans[idx] = ans[idx - 1];
12.            ans[idx - 1] = t;
13.            idx --;
14.        }
15.    }
```

6. 结果的输出

按序输出 ans[] 中的值，就是从大到小的输出答案。

```
1. // 输出结果
2. System.out.println("最大的 5 个数: ");
3. for(int i = 0; i < 5; i ++ )
4.     System.out.println(ans[i]);
```

7. 完整代码

```
1. public class Q1 {
2.     public static void main(String[] args) {
3.         // 生成 100*100 的二维数组
4.         int len = 100;
5.         double[][] a = new double[len][len];
6.
7.         // 使用 Math.random() 生成 0-1 之间的浮点数，对矩阵赋值
8.         for(int i = 0; i < len; i ++ )
9.             for(int j = 0; j < len; j ++ )
10.                a[i][j] = Math.random();
11.
12.         // 创建数组存储答案，初始化为 -1
13.         double[] ans = new double[5];
14.         for(int i = 0; i < 5; i ++ )
15.             ans[i] = -1;
```

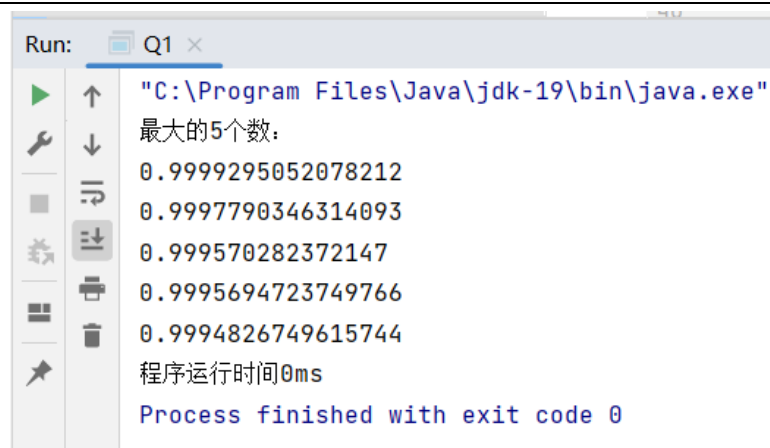
```

16.
17.     //计算程序运行时间
18.     long ST = System.currentTimeMillis();
19.     for(int i = 0; i < len; i ++ )
20.         for(int j = 0; j < len; j ++ ) {
21.             //首先将当前元素 a[i][j]与 ans[4]比较
22.             int idx = 4;
23.             if(a[i][j] > ans[idx]) {
24.                 ans[idx] = a[i][j];
25.             }
26.             //如果 ans[idx] > ans[idx - 1], 就交换二者, 同时注意
            idx 不能越界
27.             while(idx > 0 && ans[idx] > ans[idx - 1]) {
28.                 double t = ans[idx];
29.                 ans[idx] = ans[idx - 1];
30.                 ans[idx - 1] = t;
31.                 idx --;
32.             }
33.
34.         }
35.
36.     //输出结果
37.     System.out.println("最大的 5 个数: ");
38.     for(int i = 0; i < 5; i ++ )
39.         System.out.println(ans[i]);
40.
41.     long ET = System.currentTimeMillis();
42.     System.out.print("程序运行时间" + (ET - ST) + "ms");
43. }
44. }

```

8. 程序运行结果

由图 1 所示, 程序成功地将最大地 5 个数从大到小地输出了, 并且运行时间非常短, 接近 0ms。



```
Run: Q1 x
"C:\Program Files\Java\jdk-19\bin\java.exe"
最大的5个数:
0.9999295052078212
0.9997790346314093
0.999570282372147
0.9995694723749766
0.9994826749615744
程序运行时间0ms
Process finished with exit code 0
```

图 1 问题 1 程序运行结果

9. 分析算法的复杂度

由于我们遍历了一次二维数组，有两个 for 循环，所以这部分的时间复杂度为 $O(n^2)$ ，然后对于中间与 `ans[]` 进行比较的 while 循环，由于其每次最多比较 5 次，所以可以视为一个常数。故最后我们算法的时间复杂度为 $O(n^2)$ ，这比起来对数组直接进行排序是优秀很多的。

(2) 某省居民电价分三个“阶梯”：

对于用电量 50 度以内的这部分电量，电价为 0.65 元/度；

对于用电量在 51 度至 220 度之间的这部分电量，电价为 0.68 元/度；

对于用电量超过 220 度的这部分电量，电价为 0.75 元/度。

编写程序，用户从键盘输入用电量（要求在不终止程序的情况下能循环 10 次从键盘读入 10 个数字：48、52、230、90、80、89、510、60、1、10），程序输出用户应缴纳的电费。在报告中附上程序截图、运行结果截图和详细的文字说明。（25 分）

1. 从键盘输入用电量

我们创建一个输入流 `Scanner` 对象，然后通过其读取键盘的输入。

```
1. //用来读取键盘输入
2. Scanner sc = new Scanner(System.in);
```

2. 循环处理

进行 10 次循环，每次循环读入一个 `double` 类型的数，作为用户的用电量。

```
1. int cnt = 10;
2. for (int i = 0; i < cnt; i++) {
3.     System.out.print("Input:");
4.     double num = sc.nextDouble();
5.     //...
6. }
```

3. 根据用电量计算电费

- i. 如果用电量不超过 50 度，则电费为 $0.65 * \text{num}$ 。
- ii. 如果用电量超过 50 度但不超过 220 度，则电费为 50 度乘 0.65 加上超过 50 度的部分乘 0.68，即 $0.65 * 50 + (\text{num} - 50) * 0.68$ 。
- iii. 如果用电量超过 220 度，则电费为 50 度乘 0.65，加上 170 度乘 0.68，再加上超过 220 度的部分乘 0.75，即 $0.65 * 50 + 170 * 0.68 + (\text{num} - 220) * 0.75$ 。

```
1. //计算电费
2. double ans = 0;
3. if (num <= 50)
4.     ans = 0.65 * num;
5. else if (num <= 220)
6.     ans = 0.65 * 50 + (num - 50) * 0.68;
7. else
8.     ans = 0.65 * 50 + 170 * 0.68 + (num - 220) * 0.75;
```

4. 结果的输出

我们用 DecimalFormat 类使输出结果保留三位小数

```
1. //限制输出格式，保留三位小数
2. DecimalFormat df = new DecimalFormat("0.000");
3. System.out.println("用户应缴纳电费" + df.format(ans) + "元。");
```

5. 关闭输入流

在 10 个数读取完毕之后，我们需要关闭输入流，完成资源的释放。

```
1. //释放资源
2. sc.close();
```

6. 完整代码

```
1. import java.text.DecimalFormat;
2. import java.util.Scanner;
3. public class Q2 {
4.     public static void main(String[] args) {
5.         //用来读取键盘输入
6.         Scanner sc = new Scanner(System.in);
7.         int cnt = 10;
8.         for (int i = 0; i < cnt; i++) {
9.             System.out.print("Input:");
10.            double num = sc.nextDouble();
11.
12.            //计算电费
13.            double ans = 0;
14.            if (num <= 50)
15.                ans = 0.65 * num;
```

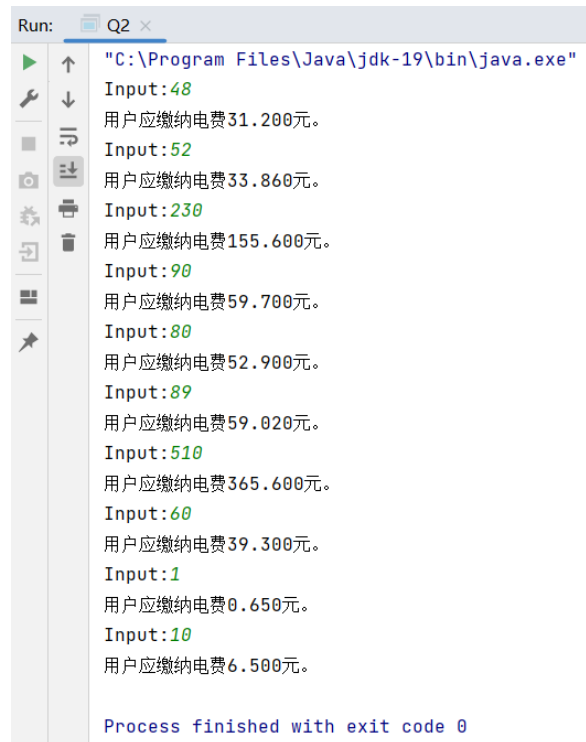
```

16.         else if (num <= 220)
17.             ans = 0.65 * 50 + (num - 50) * 0.68;
18.         else
19.             ans = 0.65 * 50 + 170 * 0.68 + (num - 220) * 0.75;
20.
21.         //限制输出格式，保留三位小数
22.         DecimalFormat df = new DecimalFormat("0.000");
23.         System.out.println("用户应缴纳电费" + df.format(ans) + "
元。");
24.     }
25.
26.     //释放资源
27.     sc.close();
28. }
29. }

```

7. 程序运行结果

由图 2 所示，程序成功计算了每个用户的电费，而且经过验证计算结果正确。



```

Run: Q2 x
"C:\Program Files\Java\jdk-19\bin\java.exe"
Input: 48
用户应缴纳电费31.200元。
Input: 52
用户应缴纳电费33.860元。
Input: 230
用户应缴纳电费155.600元。
Input: 90
用户应缴纳电费59.700元。
Input: 80
用户应缴纳电费52.900元。
Input: 89
用户应缴纳电费59.020元。
Input: 510
用户应缴纳电费365.600元。
Input: 60
用户应缴纳电费39.300元。
Input: 1
用户应缴纳电费0.650元。
Input: 10
用户应缴纳电费6.500元。
Process finished with exit code 0

```

图 2 问题 2 程序运行结果

(3) 编写程序：用户从键盘输入 1 个正整数，程序逆序打印出该正整数的各位数字（例如：某个正整数为 123456，程序输出 654321），要求在不终止程序的情况下能循环 10 次从键盘读入。在报告中附上程序截图、运行结果截图和详细的文字说明。（25 分）

1. 从键盘输入用电量

我们创建一个输入流 Scanner 对象，然后通过其读取键盘的输入。

```
1. //用来读取键盘输入
2. Scanner sc = new Scanner(System.in);
```

2. 循环处理

进行 10 次循环，每次循环读入一个 int 类型的数，作为需要进行逆序输出的正整数。

```
1. //循环处理
2. int cnt = 10;
3. for (int i = 0; i < cnt; i++) {
4.     System.out.println("Input:");
5.     int num = sc.nextInt();
6.     ...
7. }
```

3. 逆序输出

方法一

首先如果 num 为 0 的话，我们就直接输出 0。如果 num 不为 0 的话，我们就从低到高的输出 num 的每一位。具体是通过 num%10 来获得 num 的最低位，然后 num/=10 使得下一位放在最低位，直到遍历完 num 的每一位，即 num=0 时才结束。

```
1. //方法一
2. if(num == 0) {
3.     System.out.println(num);
4. }
5. //从低到高输出 num 的每一位
6. else {
7.     while(num > 0) {
8.         System.out.print(num % 10);
9.         num /= 10;
10.    }
11.    System.out.println();
12. }
```

方法二

将 num 转化为 StringBuilder，然后使用 StringBuilder 的 reverse 函数将数字逆序，最后再将逆序的结果输出。

```
1. //方法二
```

```
2. StringBuilder s = new StringBuilder("" + num);
3. //用 reverse 函数将数字逆序
4. s.reverse();
5. System.out.println(s);
```

4. 关闭输入流

在 10 个数读取完毕之后，我们需要关闭输入流，完成资源的释放。

```
1. //释放资源
2. sc.close();
```

5. 完整代码

```
1. import java.util.Scanner;
2. public class Q3 {
3.     public static void main(String[] args) {
4.         //用来读取键盘输入
5.         Scanner sc = new Scanner(System.in);
6.         //循环处理
7.         int cnt = 10;
8.         for (int i = 0; i < cnt; i++) {
9.             System.out.println("Input:");
10.            int num = sc.nextInt();
11.
12.            //方法一
13.            if(num == 0) {
14.                System.out.println(num);
15.            }
16.            //从低到高输出 num 的每一位
17.            else {
18.                while(num > 0) {
19.                    System.out.print(num % 10);
20.                    num /= 10;
21.                }
22.                System.out.println();
23.            }
24.
25.            //方法二
26.            StringBuilder s = new StringBuilder("" + num);
27.            //用 reverse 函数将数字逆序
28.            s.reverse();
29.            System.out.println(s);
30.
31.        }
```

```
32.     }  
33.     //释放资源  
34.     sc.close();  
35. }  
36. }
```

6. 程序运行结果

由图 3 可知，对于输入的每一个数，程序都将其进行逆序输出，输出结果正确。

```
Run: Q3 x  
"C:\Program Files\Java\jdk-19\bin\java.exe"  
Input:  
0  
0  
Input:  
123  
321  
Input:  
654  
456  
Input:  
789  
987  
Input:  
943727  
727349  
Input:  
417871  
178714  
Input:  
101010  
010101  
Input:  
237  
732  
Input:  
913  
319
```

图 3 问题 3 程序运行结果

+++++

其他（例如感想、建议等等）。

通过本次实验，我掌握了 JAVA 数组基本概念及应用，并且学会了如何变换数组的内容，同时我还熟练掌握数据类型、运算符、表达式和语句。

由于我已经修读过了 Java 网络编程这门科目，所以本次实验的内容我还是比较了解的，并且还会更加注意释放资源等问题，以提高代码的完整性和鲁棒性。除此之外，在有了 C++等语言基础上，Java 语言上手起来还是比较迅速的。