

Java 第二次作业

闻铭远 2000022748

6.4

```
1. package homework_2;
2.
3. /* 使用下面的方法头编写方法，反向显示一个整数：
4. public static void reverse(int number)
5. 例如：reverse(3456)返回 6543。编写一个测试程序，提示用户输入一个整数，然后显示它的
   反向数。
6. */
7.
8. import java.util.Scanner;
9.
10. public class Q6_4 {
11.     /** 将数字反向 */
12.     public static void reverse(int number) {
13.         while (number != 0) {
14.             System.out.print(number % 10); // 打印最后一位数字
15.             number = number / 10; // 去掉最后一位数字
16.         }
17.     }
18.     /** 主方法 */
19.     public static void main(String[] args) {
20.         while (true) {
21.             Scanner input = new Scanner(System.in);
22.             System.out.print("\n 请输入一个正整数: ");
23.             try { // 如果输入是整数类型，继续执行程序
24.                 int num = input.nextInt();
25.                 if (num < 0) { // 如果输入是负数，打印错误信息
26.                     System.out.print("只能输入正整数，请重新输入");
27.                     continue;
28.                 }
29.                 reverse(num); // 将数字反向
30.             } catch (Exception e) { // 如果输入不是整数类型，打印错误信息
31.                 System.out.print("只能输入正整数，请重新输入");
32.             }
33.         }
34.     }
35. }
```

```
Run: Q6_4 x
C:\APP\Java\jdk-15\bin\java.exe "-javaag
请输入一个正整数: 7726
6277
请输入一个正整数: -1001
输入错误, 请重新输入
请输入一个正整数: hello
输入错误, 请重新输入
请输入一个正整数: |
```

6.17

```
1. package homework_2;
2.
3. /* 编写一个方法, 使用下面的方法头显示 n x n 的矩阵:
4. public static void printMatrix(int n)
5. 每个元素都是随机产生的 0 或 1。编写一个测试程序, 提示用户输入 n , 显示一个 n x n 的
   矩阵。
6. */
7.
8. import java.util.Scanner;
9.
10. public class Q6_17 {
11.     /** 打印矩阵 */
12.     public static void printMatrix(int n) {
13.         int i;
14.         int j;
15.         for (i = 1; i <= n; i++) { // 循环矩阵的每行
16.             String line = "";
17.             for (j = 1; j <= n; j++) { // 循环矩阵的每列
18.                 int rand = (int)(Math.random()*2); // 随机生成 0 或 1
19.                 line = line + String.valueOf(rand) + " "; // 生成矩阵的一
   行
20.             }
21.             line = line.trim(); // 删除最右侧空白
22.             System.out.println(line); // 打印矩阵的一行
23.         }
24.     }
25.     /** 主方法 */
26.     public static void main(String[] args) {
27.         while (true) {
28.             Scanner input = new Scanner(System.in);
```

```

29.         System.out.print("\n 请输入 n 的值: ");
30.         try { // 如果输入是整数类型, 继续执行程序
31.             int n = input.nextInt();
32.             if (n <= 0) { // 如果输入是负数或 0, 打印错误信息
33.                 System.out.print("只能输入正整数, 请重新输入");
34.                 continue;
35.             }
36.             printMatrix(n); // 打印矩阵
37.         } catch (Exception e) { // 如果输入不是整数类型, 打印错误信息
38.             System.out.print("只能输入正整数, 请重新输入");
39.         }
40.     }
41. }
42. }

```

```

Run: Q6_17 x
C:\APP\Java\jdk-15\bin\java.exe
请输入n的值: 3
1 1 1
0 1 0
1 0 0
请输入n的值: 

```

7.7

```

1. package homework_2;
2.
3. /* 编写一个程序, 生成 0 和 9 之间的 100 个随机整数, 然后显示每一个数出现的次数。
4. 提示: 使用 (int)(Math.random()*10)产生 0 到 9 之间的随机整数。
5. 使用一个名为 counts 的由 10 个整数构成的数组存放 0, 1,... , 9 的个数。
6. */
7.
8. public class Q7_7 {
9.     /** 主方法 */
10.    public static void main(String[] args) {
11.        // 创建 100 个整数构成的数组
12.        int[] numbers = create_numbers();
13.        // 打印上述数组
14.        print_numbers(numbers);
15.        // 创建由 0 至 9 各自的出现次数构成的数组
16.        int[] counts = create_counts(numbers);
17.        // 打印空行










```

```

18.         System.out.println("\n");
19.         // 打印上述数组
20.         print_counts(counts);
21.     }
22.     /** 创建 0 至 9 的 100 个随机整数构成的数组 */
23.     public static int[] create_numbers() {
24.         int[] numbers = new int[100];
25.         for (int i = 0; i < 100; i++)
26.             numbers[i] = (int)(Math.random()*10);
27.         return numbers;
28.     }
29.     /** 打印 100 个整数构成的数组 */
30.     public static void print_numbers(int[] numbers) {
31.         System.out.print("100 个整数为: ");
32.         for (int i = 0; i < numbers.length; i++)
33.             System.out.print(numbers[i] + " ");
34.     }
35.     /** 创建由 0 至 9 各自的出现次数构成的数组 */
36.     public static int[] create_counts(int[] numbers) {
37.         int[] counts = new int[10];
38.         for (int i = 0; i < 100; i++)
39.             counts[numbers[i]]++;
40.         return counts;
41.     }
42.     /** 打印 counts 数组 */
43.     public static void print_counts(int[] counts) {
44.         System.out.print("0 至 9 的每个数分别出现的次数为: ");
45.         for (int i = 0; i < 10; i++)
46.             System.out.print(counts[i] + " ");
47.     }
48. }






```

Run: Q7_7 x

C:\APP\Java\jdk-15\bin\java.exe "-javaagent:C:\APP\JetBrains\IntelliJ IDEA Community Editor
100个整数为: 9 8 8 5 3 7 6 2 5 3 6 2 8 8 4 2 8 1 9 9 1 1 0 0 9 6 5 0 0 4 0 7 6 8 1 6 6 1 7 9

0至9的每个数分别出现的次数为: 14 9 8 7 5 11 12 8 14 12
Process finished with exit code 0

 Run
 TODO
 Problems
 Build
 Terminal

7.10

1. `package` homework_2;
- 2.

3. /* 编写一个方法，求出整数数组中最小元素的下标。如果这样的元素个数大于 1，则返回最小的下标。使用下面的方法头：

4. public static int indexOfSmallestElement(double[] array)

5. 编写测试程序，提示用户输入 10 个数字，调用这个方法，返回最小元素的下标，然后显示这个下标值。

6. */

7.

8. import java.util.Scanner;

9.

10. public class Q7_10 {

11. /** 获得数组中最小元素的最小下标 */

12. public static int indexOfSmallestElement(double[] array) {

13. // 初始最小值为数组的第一个元素

14. double min = array[0];

15. // 初始下标为 0

16. int indexOfMin = 0;

17. // 获得最小元素的最小下标

18. for (int i = 1; i < 10; i++) {

19. if (array[i] < min) {

20. min = array[i];

21. indexOfMin = i;

22. }

23. }

24. // 返回最小元素的最小下标

25. return indexOfMin;

26. }

27. /** 主方法 */

28. public static void main(String[] args) {

29. // 创建 10 个元素的数组

30. double[] array = new double[10];

31. // 循环输入整数，并将其对应到数组中

32. for (int i = 0; i < 10; i++) {

33. Scanner input = new Scanner(System.in);

34. System.out.println("请输入第" + String.valueOf(i + 1) + "个整数：");

35. double number = input.nextDouble();

36. array[i] = number;

37. }

38. // 打印结果

39. System.out.println("最小元素的下标值是" + indexOfSmallestElement(array));

40. }

41. }

```
Run: Q7_10 x
C:\APP\Java\jdk-15\bin\java.exe "-javaag
请输入第1个整数:
5
请输入第2个整数:
6
请输入第3个整数:
2
请输入第4个整数:
74
请输入第5个整数:
10
请输入第6个整数:
2
请输入第7个整数:
41
请输入第8个整数:
23
请输入第9个整数:
9
请输入第10个整数:
4
最小元素的下标值是2

Process finished with exit code 0
```

8.2

```
1. package homework_2;
2.
3. /* 编写一个方法，求 n x n 的 double 类型矩阵中主对角线上所有数字的和，使用下面的方法头：
4. public static double sumMajorDiagonal(double[][] m)
5. 编写一个测试程序，读取一个 4 x 4 的矩阵，然后显示它的主对角线上的所有元素的和。
6. */
7.
8. import java.util.Scanner;
9.
10. public class Q8_2 {
11.     /** 求矩阵主对角线数字之和 */
12.     public static double sumMajorDiagonal(double[][] m) {
13.         // 初始求和为 0
14.         double sum = 0;
15.         // 累加矩阵主对角线数字
16.         for (int i = 0; i < 4; i++)
```

```

17.         sum = sum + m[i][i];
18.         // 返回求和结果
19.         return sum;
20.     }
21.     /** 主方法 */
22.     public static void main(String[] args) {
23.         Scanner input = new Scanner(System.in);
24.         // 创建一个 4x4 的矩阵
25.         double[][] m = new double[4][4];
26.         System.out.println("请按行输入一个 4x4 的矩阵:");
27.         // 根据输入数字生成矩阵
28.         for (int i = 0; i < 4; i++)
29.             for (int j = 0; j < 4; j++)
30.                 m[i][j] = input.nextDouble();
31.         // 打印矩阵主对角线求和的结果
32.         System.out.println("该矩阵的主对角线上所有元素之和是
    " + sumMajorDiagonal(m));
33.     }
34. }

```

Run: Q8_2 x

C:\APP\Java\jdk-15\bin\java.exe "-jav

请按行输入一个4x4的矩阵:

1.5 2 3 4

0 1 4.5 5

7 8 9 10

5 3 2 0.1

该矩阵的主对角线上所有元素之和是11.6

Process finished with exit code 0

4: Run | TODO | 6: Problems | Build | Terminal