

基础题目

第一题：需求实现

- 随机验证码。
 - 随机生成十组六位字符组成的验证码。
 - 验证码由大小写字母、数字字符组成。
- 代码实现，效果如图所示：
- 开发提示：
 - 使用字符数组保存原始字符，利用Random类生成随机索引。
- 参考答案：

```
1
2     public class Test1 {
3         public static void main(String[] args) {
4             for (int i = 0; i < 10; i++) {
5                 String s = verifyCode();
6                 System.out.println("随机验证码:" + s);
7             }
8         }
9         public static String verifyCode() {
10             char[] arr = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
11                           'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O',
12                           'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b', 'c', 'd',
13                           'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
14                           't', 'u', 'v', 'w', 'x', 'y', 'z'};
15
16             Random random = new Random();
17             String code = "";
18             for (int i = 0; i < 6; i++) {
19                 int index = random.nextInt(arr.length);
20                 code += arr[index];
21             }
22             return code;
23         }
24     }
```

第二题：需求实现

- 键盘录入学生信息，保存到集合中。
 - 循环录入的方式，1：表示继续录入，0：表示结束录入。
 - 定义学生类，属性为姓名，年龄，使用学生对象保存录入数据。
 - 使用ArrayList集合，保存学生对象，录入结束后，遍历集合。
- 代码实现，效果如图所示：
- 参考答案：

```
1
2 public class Test2 {
3     public static void main(String[] args) {
4         Scanner scanner = new Scanner(System.in);
5         ArrayList<Student> list = new ArrayList<>();
6         while (true) {
7             System.out.println("1.录入信息 0.退出");
8             int i = scanner.nextInt();
9             switch (i) {
10                 case 1:
11                     inputStu(list, scanner);
12                     break;
13                 case 0:
14                     System.out.println("录入完毕");
15             }
16             if (i == 0){
17                 break;
18             }
19         }
20
21         for (int i = 0; i < list.size(); i++) {
22             Student student = list.get(i);
23             student.show();
24         }
25     }
26
27     private static void inputStu(ArrayList<Student> list, Scanner sc) {
28         System.out.println("请输入姓名:");
29         String name = sc.next();
30         System.out.println("请输入年龄:");
31         int age = sc.nextInt();
32         Student student = new Student(name, age);
33         list.add(student);
34     }
35 }
36
37 class Student{
38     private String name;
39     private int age;
40     public Student() {
41     }
42     public Student(String name, int age) {
43         this.name = name;
44         this.age = age;
45     }
46     public String getName() {
47         return name;
48     }
49     public void setName(String name) {
50         this.name = name;
51     }
52     public int getAge() {
53         return age;
54     }
55     public void setAge(int age) {
56         this.age = age;
57     }
58 }
```

```

58         public void show() {
59             system.out.println( "学生姓名=" + name + ", 年龄=" + age);
60         }
61     }

```

第三题：需求实现

- 集合工具类。
 - 定义findIndex方法，在某集合中，查找某元素，返回第一次出现的索引。
 - 定义replace方法，将某集合中的某元素，全部替换为新元素。
- 代码实现，效果如图所示：
- 参考答案：

```

1
2     public class Test3 {
3         public static int findIndex(List<Integer> list, int i){
4             int index = -1;
5             for (int j = 0; j < list.size(); j++) {
6                 if (list.get(j) == i) {
7                     index = j;
8                     break;
9                 }
10            }
11            return index;
12        }
13        public static void replace(List<Integer> list,Integer
oldValue,Integer newValue){
14            for (int i = 0; i < list.size(); i++) {
15                if (list.get(i) == oldValue) {
16                    list.set(i, newValue);
17                }
18            }
19        }
20    }

```

第四题：需求实现

- 统计数字出现次数。
 - 定义getNumList方法，随机生成100个数字，数字范围从1到10。
 - 定义printCount方法，统计每个数字出现的次数并打印到控制台。
- 代码实现，部分效果如图所示：
- 参考答案：

```

1
2     public class Test4 {
3         public static void main(String[] args) {
4             ArrayList<Integer> numList = getNumList();
5             // 统计字符数组中字母出现次数
6             printCount(numList);
7         }
8         public static void printCount(ArrayList<Integer> list) {

```

```

9         int[] count = new int[10];
10        // 对应保存数字出现的次数
11        for (int i = 0; i < list.size(); i++) {
12            int c = list.get(i);
13            count[c-1]++;
14        }
15        // 打印数字和次数
16        for (int i = 0 ; i < count.length; i++) {
17            System.out.println("数字:"+(i+1) + "--" + count[i]+"次");
18        }
19    }
20    public static ArrayList<Integer> getNumList() {
21        ArrayList<Integer> list = new ArrayList<>();
22        Random r = new Random();
23        for (int i = 0; i < 100; i++) {
24            int x = r.nextInt(10) + 1;
25            list.add(x);
26        }
27        return list;
28    }
29 }

```

第五题：需求实现

- 模拟统计班级考试分数分布情况，分别统计100-80，79-60，59-40，39-0各个阶段的人数。
 - 定义getScoreList方法，随机生成50个数字，数字范围从0到100。
 - 定义countScore方法，统计各个阶段的分数个数。
 - 定义printCount方法，打印各个阶段的统计结果。
- 代码实现，效果如图所示：
- 参考答案：

```

1
2     public class Test5 {
3         public static void main(String[] args) {
4             // 获取随机分数
5             ArrayList<Integer> scoreList = getScoreList();
6             // 定义计数的变量
7             ArrayList<Integer> countList = countScore(scoreList);
8             // 打印统计结果
9             printCount(countList);
10        }
11
12        private static void printCount(ArrayList<Integer> countList) {
13            int start = 100;
14            int end = 80;
15            for (int i = 0; i < countList.size(); i++) {
16                Integer integer = countList.get(i);
17                System.out.println(start + "\t分 --" + end + " \t分:" +
integer+"人");
18                if (i == 0){
19                    start-=21;
20                    end -=20;
21                }else if (i == countList.size()-2){

```

```

22         start-=20;
23         end-=40;
24     }else {
25         start -= 20;
26         end -= 20;
27
28     }
29 }
30 }
31 public static ArrayList<Integer> countScore(ArrayList<Integer>
scoreList) {
32     ArrayList<Integer> countList = new ArrayList<>();
33     int count100 = 0;
34     int count79 = 0;
35     int count59 = 0;
36     int count39 = 0;
37
38     for (int i = 0; i < scoreList.size(); i++) {
39         Integer score = scoreList.get(i);
40         if (score <= 100 && score >= 80) {
41             count100++;
42         } else if (score <= 79 && score >= 60) {
43             count79++;
44         } else if (score <= 59 && score >= 40) {
45             count59++;
46         } else {
47             count39++;
48         }
49     }
50
51     countList.add(count100);
52     countList.add(count79);
53     countList.add(count59);
54     countList.add(count39);
55
56     return countList;
57 }
58
59
60 public static ArrayList<Integer> getScoreList() {
61     ArrayList<Integer> list = new ArrayList<>();
62     Random r = new Random();
63     for (int i = 0; i < 50; i++) {
64         int x = r.nextInt(100);
65         list.add(x);
66     }
67     return list;
68 }
69 }

```

第六题：需求实现

- 随机生成n张扑克牌。
- 代码实现，效果如图所示：
- 开发提示：
 - 使用集合保存所有的扑克牌对象。

- 从所有牌中，随机移除n张牌，保存到新集合。
- 判断n的值，不能超越一副扑克牌的范围。

- 参考答案：

```
1
2     public class Test6 {
3         public static void main(String[] args) {
4             int n = 5;
5             ArrayList<Card> cards = randomCard(n);
6
7             if (cards != null) {
8                 System.out.println("随机"+ n +"张牌:" );
9                 for (int i = 0; i < cards.size(); i++) {
10                     Card card = cards.get(i);
11                     card.showCard();
12                 }
13             }else {
14                 System.out.println(n+"超越范围,无法获取牌" );
15             }
16
17             System.out.println();
18             System.out.println();
19             int n2 = 55;
20             ArrayList<Card> cards2 = randomCard(n2);
21
22             if (cards2 != null) {
23                 System.out.println("随机"+ n2 +"张牌:" );
24                 for (int i = 0; i < cards.size(); i++) {
25                     Card card = cards.get(i);
26                     card.showCard();
27                 }
28             }else {
29                 System.out.println("随机"+ n2 +"张牌:\r\n超越范围,无法获取" );
30             }
31         }
32         public static ArrayList<Card> randomCard(int n) {
33             if (n > 54 || n < 0)
34                 return null;
35
36             ArrayList<Card> rList = new ArrayList<>();
37             ArrayList<Card> cards = allCard();
38
39             Random r = new Random();
40             for (int i = 0; i < n; i++) {
41                 int index = r.nextInt(cards.size());
42                 Card rCard = cards.remove(index);
43                 rList.add(rCard);
44             }
45             return rList;
46         }
47
48         public static ArrayList<Card> allCard() {
49             ArrayList<Card> allList = new ArrayList<>();
50             // 花色数组
51             String[] hs = {"黑桃", "红桃", "梅花", "方片"};
52             // 点数数组
```

```

53         String[] ds = {"A", "2", "3", "4", "5", "6", "7", "8", "9",
54         "10", "J", "Q", "K"};
55         for (int H = 0; H < hs.length; H++) {
56             for (int d = 0; d < ds.length; d++) {
57                 Card card = new Card(hs[H], ds[d]);
58                 // 添加到集合
59                 allList.add(card);
60             }
61         }
62         return allList;
63     }
64 }
65
66 class Card {
67     private String ds; // 点数
68     private String hs; // 花色
69     public Card(String ds, String hs) {
70         this.ds = ds;
71         this.hs = hs;
72     }
73     public void showCard() {
74         System.out.print(ds + hs + " ");
75     }
76 }

```

第七题：需求实现

- 自定义MyList类，实现存取元素的功能。
 - 定义add方法，可以保存元素，添加MyList尾部。
 - 定义remove方法，可以获取到最后添加的元素，并从MyList中移除该元素。
 - 定义show方法，可以展示MyList中的元素。
- 代码实现，效果如图所示：
- 参考答案：

```

1
2     public class Test7 {
3         public static void main(String[] args) {
4             MyList myList = new MyList();
5             for (int i = 0; i < 3; i++) {
6                 myList.add(i);
7             }
8             System.out.println("添加元素后:");
9             myList.show();
10
11             Integer remove = myList.remove();
12             System.out.println("获取元素:");
13             System.out.println(remove);
14             System.out.println("获取元素后:");
15             myList.show();
16         }
17     }
18
19     class MyList {

```

```
20     ArrayList<Integer> m1 = new ArrayList<>();
21
22     public void add(Integer i) {
23         m1.add(i);
24     }
25     public Integer remove() {
26         Integer remove = m1.remove(m1.size() - 1);
27         return remove;
28     }
29
30     public void show(){
31         System.out.println(m1);
32     }
33 }
```