

第2讲-Java编程基础7综合练习

练习一：飞机票

需求:

机票价格按照淡季旺季、头等舱和经济舱收费、输入机票原价、月份和头等舱或经济舱。

按照如下规则计算机票价格：旺季（5-10月）头等舱9折，经济舱8.5折，淡季（11月到来年4月）头等舱7折，经济舱6.5折。

代码示例：

```
package com.lxm.test;

import java.util.Scanner;

public class Test1 {
    public static void main(String[] args) {
        /* 机票价格按照淡季旺季、头等舱和经济舱收费、输入机票原价、月份和头等舱或经济舱。
        按照如下规则计算机票价格：旺季（5-10月）头等舱9折，经济舱8.5折，淡季（11月到
        来年4月）头等舱7折，经济舱6.5折。*/

        //分析：
        //1.键盘录入机票原价、月份、头等舱或经济舱
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入机票的原价");
        int ticket = sc.nextInt();
        System.out.println("请输入当前的月份");
        int month = sc.nextInt();
        System.out.println("请输入当前购买的舱位 0 头等舱 1 经济舱");
        int seat = sc.nextInt();
        //2.先判断月份是旺季还是淡季
        //ctrl + alt + M 自动抽取方法
        if (month >= 5 && month <= 10) {
            //旺季 //3.继续判断当前机票是经济舱还是头等舱
            //ticket = getPrice(ticket, seat, 0.9, 0.85);
            ticket = getTicket(ticket, seat, 0.9, 0.85);
        } else if ((month >= 1 && month <= 4) || (month >= 11 && month <= 12))
        {
            //淡季
            //ticket = getPrice(ticket, seat, 0.7, 0.65);
            ticket = getTicket(ticket, seat, 0.7, 0.65);
        } else {
            //表示键盘录入的月份是一个非法数据
            System.out.println("键盘录入的月份不合法");
        }
    }
}
```

```

        System.out.println(ticket);
    }

    public static int getTicket(int ticket, int seat, double v, double v2) {
        if (seat == 0) {
            //头等舱
            ticket = (int) (ticket * v);
        } else if (seat == 1) {
            //经济舱
            ticket = (int) (ticket * v2);
        } else {
            System.out.println("没有这个舱位");
        }
        return ticket;
    }

    //1.我要干嘛？ 根据舱位和折扣来计算最终的票价
    //2.我干这件事，需要什么才能完成？ 原价 舱位 头等舱的折扣 经济舱的折扣
    //3.方法的调用处是否需要继续使用这个结果 需要
    /* public static int getPrice(int ticket, int seat, double v0, double v1)
    {
        if (seat == 0) {
            //头等舱
            ticket = (int) (ticket * v0);
        } else if (seat == 1) {
            //经济舱
            ticket = (int) (ticket * v1);
        } else {
            System.out.println("没有这个舱位");
        }
        return ticket;
    }
    */
}

```

练习二：打印素数

判断101~200之间有多少个素数，并输出所有素数。

备注：素数就是质数

代码示例：

```

package com.lxm.test;

public class Test2 {
    public static void main(String[] args) {
        //判断 101 ~ 200 之间有多少个素数，并打印所有素数
    }
}

```

```

//思路一： 2 ~ 99
//定义变量i , 赋值100
//判断i是否为质数

//定义一个变量用来统计有多少个质数
int count = 0;
//外循环：遍历101~200这个范围，依次得到这个范围内的每一个数字
for (int i = 101; i <= 200; i++) {
    //i 依次表示循环中的每一个数字
    //继续判断i是否为一个质数
    boolean flag = true;
    //内循环：判断当前数字是否为一个质数。
    for (int j = 2; j < i; j++) {
        //j 表示2~99之间的每一个数字
        if(i % j == 0){
            flag = false;
            //跳出单层循环，内循环
            break;
        }
    }
    if(flag){
        System.out.println("当前数字"+i+"是质数");
        count++;
    }
}

System.out.println("一共有" + count + "个质数");

/* int i = 7;
boolean flag = true;
for (int j = 2; j < i; j++) {
    //j 表示2~99之间的每一个数字
    if(i % j == 0){
        flag = false;
        break;
    }
}
if(flag){
    System.out.println("当前数字是质数");
}else{
    System.out.println("当前数字不是一个质数");
}*/
}
}

```

练习三：验证码

需求：

定义方法实现随机产生一个5位的验证码

验证码格式:

长度为5

前四位是大写字母或者小写字母

最后一位是数字

代码示例:

```
package com.lxm.test;

import java.util.Random;

public class Test3 {
    public static void main(String[] args) {
        /*    需求:
            定义方法实现随机产生一个5位的验证码
            验证码格式:
            长度为5
            前四位是大写字母或者小写字母
            最后一位是数字
        */

        //方法:
        //在以后如果我们要在一堆没有什么规律的数据中随机抽取
        //可以先把这些数据放到数组当中
        //再随机抽取一个索引

        //分析:
        //1.大写字母和小写字母都放到数组当中
        char[] chs = new char[52];
        for (int i = 0; i < chs.length; i++) {
            //ASCII码表
            if(i <= 25){
                //添加小写字母
                chs[i] = (char)(97 + i);
            }else{//27
                //添加大写字母
                // A --- 65
                chs[i] = (char)(65 + i - 26);
            }
        }

        //定义一个字符串类型的变量,用来记录最终的结果
        String result = "";

        //2.随机抽取4次
        //随机抽取数组中的索引
        Random r = new Random();
```

```

        for (int i = 0; i < 4; i++) {
            int randomIndex = r.nextInt(chs.length);
            //利用随机索引, 获取对应的元素
            //System.out.println(chs[randomIndex]);
            result = result + chs[randomIndex];
        }
        //System.out.println(result);
        //3.随机抽取一个数字0~9
        int number = r.nextInt(10);
        //生成最终的结果
        result = result + number;

        //打印最终结果
        System.out.println(result);
    }
}

```

练习四：复制数组

需求：

把一个数组中的元素复制到另一个新数组中去。

代码示例：

```

package com.lxm.test;

public class Test4 {
    public static void main(String[] args) {
        /* 需求：
        把一个数组中的元素复制到另一个新数组中去。*/

        //分析：
        //1.定义一个老数组并存储一些元素
        int[] arr = {1,2,3,4,5};
        //2.定义一个新数组的长度跟老数组一致
        int[] newArr = new int[arr.length];
        //3.遍历老数组, 得到老数组中的每一个元素, 依次存入到新数组当中
        for (int i = 0; i < arr.length; i++) {
            //i 表示老数组中的索引。新数组中的每一个索引
            //arr[i] 表示老数组中的元素
            newArr[i] = arr[i];
        }

        //4.新数组中已经存满元素了
        for (int i = 0; i < newArr.length; i++) {
            System.out.println(newArr[i]);
        }
    }
}

```

```
}  
  
}  
  
}
```

练习五：评委打分

需求：

在唱歌比赛中，有6名评委给选手打分，分数范围是[0 - 100]之间的整数。选手的最后得分为：去掉最高分、最低分后的4个评委的平均分，请完成上述过程并计算出选手的得分。

代码示例：

```
package com.lxm.test;  
  
import java.util.Scanner;  
  
public class Test5 {  
    public static void main(String[] args) {  
        //在唱歌比赛中，有6名评委给选手打分，分数范围是[0 - 100]之间的整数。  
        // 选手的最后得分为：去掉最高分、最低分后的4个评委的平均分，请完成上述过程并  
        计算出选手的得分。  
  
        //分析：  
        //1.定义一个数组，用来存储6名评委的打分 (0~100)  
        int[] scoreArr = getScores();  
        for (int i = 0; i < scoreArr.length; i++) {  
            System.out.println(scoreArr[i]);  
        }  
        //2.求出数组中的最大值  
        int max = getMax(scoreArr);  
        //3.求出数组中的最小值  
        int min = getMin(scoreArr);  
        //4.求出数组中6个分数的总和  
        int sum = getSum(scoreArr);  
        //5. (总和 - 最大值 - 最小值) / 4  
        int avg = (sum - max - min) / (scoreArr.length - 2);  
        //6.打印结果  
        System.out.println("选手的最终得分为: " + avg);  
    }  
  
    public static int getSum(int[] scoreArr){  
        int sum = 0;  
        for (int i = 0; i < scoreArr.length; i++) {  
            sum = sum + scoreArr[i];  
        }  
    }  
}
```

```

    }
    return sum;
}

//求数组的最大值
public static int getMax(int[] scoreArr){
    int max = scoreArr[0];
    for (int i = 1; i < scoreArr.length; i++) {
        if(scoreArr[i] > max){
            max = scoreArr[i];
        }
    }
    return max;
}

//求数组的最小值
public static int getMin(int[] scoreArr){
    int min = scoreArr[0];
    for (int i = 1; i < scoreArr.length; i++) {
        if(scoreArr[i] < min){
            min = scoreArr[i];
        }
    }
    return min;
}

//1.我要干嘛？ 定义一个数组，用来存储6名评委的打分（0~100）
//2.我需要什么？ 都不需要
//3.干完了这件事情，是否需要返回？ 必须返回
public static int[] getScores(){
    //定义数组
    int[] scores = new int[6];
    //使用键盘录入的形式，输入分数：0~100
    Scanner sc = new Scanner(System.in);
    for (int i = 0; i < scores.length; ) {
        System.out.println("请输入评委的打分");
        int score = sc.nextInt();//100
        if(score >=0 && score<= 100){
            scores[i] = score;
            i++;
        }else{
            System.out.println("成绩超出了范围,继续录入, 当前的i为: " + i);
        }
    }
    return scores;
}
}

```

练习六：数字加密

需求：

某系统的数字密码（大于0），比如1983，采用加密方式进行传输。

规则如下：

先得到每位数，然后每位数都加上5，再对10求余，最后将所有数字反转，得到一串新数。

举例：

	1	9	8	3
+5	6	14	13	8
%10	6	4	3	8
反转	8	3	4	6
加密后的结果就是：	8	3	4	6

代码示例：

```
package com.lxm.test;

public class Test6 {
    public static void main(String[] args) {
        /*
        某系统的数字密码（大于0）。比如1983，采用加密方式进行传输，
        规则如下：
            每位数加上5
            再对10求余，
            最后将所有数字反转，
            得到一串新数。
        */

        //分析：
        //1.把整数里面的每一位放到数组当中
        int[] arr = {1, 9, 8, 3};
        //2.加密
        //每位数加上5
        for (int i = 0; i < arr.length; i++) {
            arr[i] = arr[i] + 5;
        }
        //再对10求余，
        for (int i = 0; i < arr.length; i++) {
            arr[i] = arr[i] % 10;
        }
        //将所有数字反转
    }
}
```



```

        for (int i = 0, j = arr.length - 1; i < j; i++, j--) {
            int temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
        //8 3 4 6 --> 8346
        //3.把数组里面的每一个数字进行拼接，变成加密之后的结果
        int number = 0;
        for (int i = 0; i < arr.length; i++) {
            number = number * 10 + arr[i];
        }
        System.out.println(number);
    }
}

```

练习六扩展：

```

package com.lxm.test;

public class Test7 {
    public static void main(String[] args) {
        //需求：
        //把整数上的每一位都添加到数组当中
        //反向推导

        //1.计算出数组的长度
        int number = 12345;
        //定义一个变量临时记录number的值，就是为了第三步的时候再次使用
        int temp = number;
        //定义一个变量进行统计
        int count = 0;
        while(number != 0){
            //每一次循环就去掉右边的一个数字
            number = number / 10;
            //去掉一位计数器就自增一次。
            count++;
        }
        //2.定义数组
        //动态初始化
        int[] arr = new int[count];
        //3.把整数上的每一位都添加到数组当中
        int index = arr.length - 1;
        while(temp != 0){//12345
            //获取temp里面的每一位数组
            int ge = temp % 10;
            //再去掉右边的那位数字
            temp = temp / 10;
        }
    }
}

```

```

        //把当前获取到的个位添加到数组当中
        arr[index] = ge;
        index--;
    }
    //验证结果 1 2 3 4 5
    for (int i = 0; i < arr.length; i++) {
        System.out.print(arr[i] + " ");
    }
}
}

```

练习七：数字解密

把上一题加密之后的数据进行解密

代码示例：

```

package com.lxm.test;

public class Test8 {
    public static void main(String[] args) {
        /*某系统的数字密码（大于0）。比如1983，采用加密方式进行传输，
        规则如下：
            每位数加上5
            再对10求余，
            最后将所有数字反转，
            得到一串新数。
        按照以上规则进行解密：
            比如1983加密之后变成8346，解密之后变成1983
        */

        //1.定义数组记录解密之后的结果
        int[] arr = {8, 3, 4, 6};
        //2.反转
        for (int i = 0, j = arr.length - 1; i < j; i++, j--) {
            int temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
        //3.由于加密是通过对10取余的方式进行获取的
        //所以在解密的时候就需要判断，0~4之间+10 5~9数字不变
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] >= 0 && arr[i] <= 4) {
                arr[i] = arr[i] + 10;
            }
        }
        //4.每一位减5
        for (int i = 0; i < arr.length; i++) {

```

```

        arr[i] = arr[i] - 5;
    }
    //5.获取数组里面的每一位数字拼接成最终的结果
    int number = 0;
    for (int i = 0; i < arr.length; i++) {
        number = number * 10 + arr[i];
    }
    System.out.println(number);
}
}

```

练习八：抽奖

需求：

一个大V直播抽奖，奖品是现金红包，分别有{2, 588, 888, 1000, 10000}五个奖金。请使用代码模拟抽奖，打印出每个奖项，奖项的出现顺序要随机且不重复。打印效果如下：（随机顺序，不一定是下面的顺序）

```

888元的奖金被抽出
588元的奖金被抽出
10000元的奖金被抽出
1000元的奖金被抽出
2元的奖金被抽出

```

解法一：

```

package com.lxm.test;

import java.util.Random;

public class Test9 {
    public static void main(String[] args) {
        /* 需求：
        一个大V直播抽奖，奖品是现金红包，分别有{2, 588, 888, 1000, 10000}五个奖
        金。
        请使用代码模拟抽奖，打印出每个奖项，奖项的出现顺序要随机且不重复。
        打印效果如下：（随机顺序，不一定是下面的顺序）
        888元的奖金被抽出
        588元的奖金被抽出
        10000元的奖金被抽出
        1000元的奖金被抽出
        */
    }
}

```

```

        2元的奖金被抽出
    */

    //分析:
    //1.定义数组表示奖池
    int[] arr = {2, 588, 888, 1000, 10000};
    //2.定义新数组用于存储抽奖的结果
    int[] newArr = new int[arr.length];
    //3.抽奖
    Random r = new Random();
    //因为有5个奖项, 所以这里要循环5次
    for (int i = 0; i < 5; ) {
        //获取随机索引
        int randomIndex = r.nextInt(arr.length);
        //获取奖项
        int prize = arr[randomIndex];
        //判断当前的奖项是否存在, 如果存在则重新抽取, 如果不存在, 就表示是有效奖

        boolean flag = contains(newArr, prize);
        if(!flag){
            //把当前抽取到的奖项添加到newArr当中
            newArr[i] = prize;
            //添加完毕之后, 移动索引
            i++;
        }
    }
    //4.遍历newArr
    for (int i = 0; i < newArr.length; i++) {
        System.out.println(newArr[i]);
    }

}

//判断prize在数组当中是否存在
//存在: true
//不存在: false
public static boolean contains(int[] arr,int prize){
    for (int i = 0; i < arr.length; i++) {
        if(arr[i] == prize){
            return true;
        }
    }
    return false;
}

}

```

解法二：

```
package com.lxm.test;

import java.util.Random;

public class Test10 {
    public static void main(String[] args) {
        /* 需求：
        一个大V直播抽奖，奖品是现金红包，分别有{2, 588, 888, 1000, 10000}五个奖
        金。

        请使用代码模拟抽奖，打印出每个奖项，奖项的出现顺序要随机且不重复。
        打印效果如下：（随机顺序，不一定是下面的顺序）
            888元的奖金被抽出
            588元的奖金被抽出
            10000元的奖金被抽出
            1000元的奖金被抽出
            2元的奖金被抽出
        */

        //1.把奖池里面的所有奖项打乱顺序
        int[] arr = {2, 588, 888, 1000, 10000};
        Random r = new Random();
        for (int i = 0; i < arr.length; i++) {
            //获取随机索引
            int randomIndex = r.nextInt(arr.length);
            //拿着i跟随机索引randomIndex上的值进行交换
            int temp = arr[i];
            arr[i] = arr[randomIndex];
            arr[randomIndex] = temp;
        }
        //2.遍历奖池,从0索引开始获取每一个奖项
        for (int i = 0; i < arr.length; i++) {
            System.out.println(arr[i]);
        }
    }
}
```

练习九：双色球

投注号码由6个红色球号码和1个蓝色球号码组成。红色球号码从1—33中选择；蓝色球号码从1—16中选择。

双色球中奖条件和奖金表

奖等	中奖条件		中奖说明	单注奖金分配
	红球	蓝球		
一等奖	●●●●●●	●	中6+1	最高1000万
二等奖	●●●●●●		中6+0	最高500万
三等奖	●●●●●	●	中5+1	3000元
四等奖	●●●●●		中5+0	200元
	●●●●●	●	中4+1	
五等奖	●●●●		中4+0	10元
	●●●●	●	中3+1	
六等奖	●●	●	中2+1	5元
	●	●	中1+1	
		●	中0+1	

代码示例:

```
package com.lxm.test;

import java.util.Random;
import java.util.Scanner;

public class Test11 {
    public static void main(String[] args) {
        //1.生成中奖号码
        int[] arr = createNumber(); // 123456 7

        System.out.println("=====");
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }

        System.out.println("=====");

        //2.用户输入彩票号码 (红球 + 蓝球) //654321
        int[] userInputArr = userInputNumber();

        //3.判断用户的中奖情况
        //红球 蓝球
```

```

int redCount = 0;
int blueCount = 0;

//判断红球
for (int i = 0; i < userInputArr.length - 1; i++) {
    int redNumber = userInputArr[i];
    for (int j = 0; j < arr.length - 1; j++) {
        if(redNumber == arr[j]){
            redCount++;
            //如果找到了，那么后面的数字就没有必要继续比较了
            //跳出内循环，继续判断下一个红球号码是否中奖
            break;
        }
    }
}

//判断蓝球
int blueNumber = userInputArr[userInputArr.length-1];
if(blueNumber == arr[arr.length - 1]){
    blueCount++;
}

//根据红球的个数以及蓝球的个数来判断中奖情况
if(redCount == 6 && blueCount == 1){
    System.out.println("恭喜你，中奖1000万");
}else if(redCount == 6 && blueCount == 0){
    System.out.println("恭喜你，中奖500万");
}else if(redCount == 5 && blueCount == 1){
    System.out.println("恭喜你，中奖3000");
}else if((redCount == 5 && blueCount == 0) || (redCount == 4 &&
blueCount == 1)){
    System.out.println("恭喜你，中奖200");
}else if((redCount == 4 && blueCount == 0) || (redCount == 3 &&
blueCount == 1)){
    System.out.println("恭喜你，中奖10");
}else if((redCount == 2 && blueCount == 1) || (redCount == 1 &&
blueCount == 1) || (redCount == 0 && blueCount == 1)){
    System.out.println("恭喜你，中奖5");
}else{
    System.out.println("谢谢参与，谢谢惠顾");
}

}

public static int[] userInputNumber() {
    //1.创建数组用于添加用户购买的彩票号码
    //6个红球 1个蓝球 数组长度：7
    int[] arr = new int[7];

    //2.利用键盘录入让用户输入
    Scanner sc = new Scanner(System.in);
    //让用户输入红球号码
    for (int i = 0; i < 6; ) {

```

```

        System.out.println("请输入第" + (i + 1) + "个红球号码");
        int redNumber = sc.nextInt();
        //redNumber 在1~33 唯一不重复
        if (redNumber >= 1 && redNumber <= 33) {
            boolean flag = contains(arr, redNumber);
            if (!flag) {
                //不存在
                //有效的, 可以添加到数组当中
                arr[i] = redNumber;
                i++;
            } else {
                //存在
                System.out.println("当前红球号码已经存在, 请重新输入");
            }
        } else {
            System.out.println("当前红球号码超出范围");
        }
    }

    //让用户输入篮球号码
    System.out.println("请输入篮球号码");
    //1~16
    while (true) {
        int blueNumber = sc.nextInt();
        if (blueNumber >= 1 && blueNumber <= 16) {
            arr[arr.length - 1] = blueNumber;
            break;
        } else {
            System.out.println("当前篮球号码超出范围");
        }
    }
    return arr;
}

public static int[] createNumber() {
    //1.创建数组用于添加中奖号码
    //6个红球 1个蓝球 数组长度: 7
    int[] arr = new int[7];

    //2.随机生成号码并添加到数组当中
    //红球: 不能重复的 1 2 3 4 5 6
    //蓝球: 可以跟红球号码重复 5

    //生成红球号码并添加到数组当中
    Random r = new Random();
    for (int i = 0; i < 6; ) {
        //获取红球号码
        int redNumber = r.nextInt(33) + 1;
        boolean flag = contains(arr, redNumber);
        if (!flag) {
            //把红球号码添加到数组当中

```



```

        arr[i] = redNumber;
        i++;
    }
}

//生成篮球号码并添加到数组当中
int blueNumber = r.nextInt(16) + 1;
arr[arr.length - 1] = blueNumber;
return arr;
}

//用于判断数组在数组中是否存在
public static boolean contains(int[] arr, int number) {
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] == number) {
            return true;
        }
    }
    return false;
}
}

```