

语言基础第七天：

回顾：

1. 数组的复制：

```
1) System.arraycopy(a, 1, b, 0, 4);
2) int[] b = Arrays.copyOf(a, 6);
   a = Arrays.copyOf(a, a.length+1);
```

2. 方法：

- 封装一段特定的业务逻辑功能、只干一件事、反复多次调用、减少重复、有利于维护
- 何时用：只要是一个独立的业务功能，就要把它封装到一个方法中

3. 方法的定义：

```
修饰词 返回值类型 方法名(参数列表){
    方法体-----具体的业务逻辑功能实现
}
```

4. 方法的调用：

- 无返回值：方法名(有参传参);
- 有返回值：数据类型 变量 = 方法名(有参传参); //-----最常规的调用方式
System.out.println(方法名(有参传参)); //返回值为基本类型时可以这样调

5. return：

- return 值; //1)结束方法的执行 2)返回结果给调用方
- return; //1)结束方法的执行

精华笔记：

1. 案例：-----培养编程思维能力

笔记：

1. 案例：-----培养编程思维能力

```
package day07;

import java.util.Scanner;

/**
 * 1. 需求:猜数字小游戏
 * 训练目标: while(true)自造死循环+break
 */
public class Guessing {
```

```

public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    int num = (int)(Math.random()*1000+1); //1到1000
    System.out.println(num); //作弊

    while(true){ //自造死循环
        System.out.println("猜吧!");
        int guess = scan.nextInt();
        if(guess>num){
            System.out.println("猜大了");
        }else if(guess<num){
            System.out.println("猜小了");
        }else{
            System.out.println("恭喜你猜对了");
            break; //跳出循环
        }
    }
}
}

```

```

package day07;

import java.util.Random;

/**
 * 2. 需求：随机生成N位验证码(大写字母、小写字母、数字)
 * 训练目标：数组+方法+随机数
 */
public class VerificationCode {
    public static void main(String[] args) {
        String code = generateVeriCode(6);
        System.out.println("验证码:" + code);
    }

    /** 生成N位验证码 */
    public static String generateVeriCode(int len){
        String code = ""; //验证码
        Random rand = new Random(); //随机数对象
        char[] chs = {'a','b','c','d','e','f','g','h','i','j','k',
            'l','m','n','o','p','q','r','s','t','u','v',
            'w','x','y','z','A','B','C','D','E','F','G',
            'H','I','J','K','L','M','N','O','P','Q','R',
            'S','T','U','V','W','X','Y','Z','0','1','2',
            '3','4','5','6','7','8','9'}; //验证码字符可选范围(共62个)
        for(int i=1;i<=len;i++) { //len次
            int index = rand.nextInt(chs.length); //随机下标(0到61)
            code += chs[index]; //根据随机下标获取对应字符并拼接到code中
        }
    }

    /**
     *
     * code=""
     * i=1 index=0 code="a"
     * i=2 index=61 code="a9"
     * i=3 index=25 code="a9z"
     * i=4 index=28 code="a9zC"
     * i=5
     */
}

```

```

        */
        return code;
    }
}

```

```

package day07;

/**
 * 3.需求:找到2到100之间的所有素数(质数)
 *      素数为只能被1和它本身整除
 *      训练目标:通过boolean的flag打标记(3步)
 */
public class PrimeNumber {
    public static void main(String[] args) {
        //带数(2/3/4/5/6/7/8)
        for(int num=2;num<=100;num++){
            boolean flag = true; //假设每个num都是true
            for(int i=2;i<=num/2;i++){
                if(num%i==0){
                    flag = false;
                    break;
                }
            }
            if(flag){
                System.out.print(num+"\t");
            }
        }

        /*
        // 7%2/3/4/5/6, 但凡有1个为0的, 就能说明它不是素数, 只有都不为0的, 才是素数
        int num = 7; //7,8,9,11
        boolean flag = true; //1)假设num是素数
        for(int i=2;i<=num/2;i++){ //i=2/3
            if(num%i==0){
                flag = false; //2)修改为不是素数
                break;
            }
        }
        if(flag){ //3)判断flag标记
            System.out.println(num+"是素数");
        }else{
            System.out.println(num+"不是素数");
        }
        */
    }
}

```

```

package day07;

import java.util.Scanner;

/**
 * 4.需求:机票价格按照季节(淡季、旺季)、舱位(头等舱、商务舱、经济舱)收费
 *      要求:

```

```

*      输入机票原价、月份和舱位，实现不同的折扣
*      ---旺季(5月到10月)时，头等舱9折，商务舱85折，经济舱8折
*      ---淡季(11月到次年4月)时，头等舱7折，商务舱65折，经济舱6折
*      训练目标：分支结构
*/
public class CalAirPrice {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("请输入机票原价:");
        double price = scan.nextDouble();
        System.out.println("请输入月份:");
        int month = scan.nextInt();
        System.out.println("请选择舱位：1.头等舱 2.商务舱 3.经济舱");
        int type = scan.nextInt();

        double finalPrice = calFinalPrice(price, month, type); //计算折扣后金额
        if(finalPrice!=-1){ //数据合法
            System.out.println("机票的最终价格为:"+finalPrice);
        }
    }

    /**
     * 根据原价、月份、舱位，计算飞机票的最终价格
     */
    public static double calFinalPrice(double price,int month,int type){
        double finalPrice = 0.0; //最终价格
        //只要数据输入错误，都统一返回-1
        if(price<0){
            System.out.println("机票原价输入错误");
            return -1;
        }
        if(month<1 || month>12){
            System.out.println("月份输入错误");
            return -1;
        }
        if(type<1 || type>3){
            System.out.println("舱位输入错误");
            return -1;
        }

        //程序能走到这，说明数据一定是合法的
        if(month>=5 && month<=10){ //旺季
            switch(type){ //根据舱位类型做不同折扣
                case 1:
                    finalPrice = price*0.9;
                    break;
                case 2:
                    finalPrice = price*0.85;
                    break;
                case 3:
                    finalPrice = price*0.8;
                    break;
            }
        }else{ //淡季
            if(type==1){

```

```

        finalPrice = price*0.7;
    }else if(type==2){
        finalPrice = price*0.65;
    }else{
        finalPrice = price*0.6;
    }
}
return finalPrice;
}
}

```

```

package day07;

import java.util.Scanner;

/**
 * 5.需求:<<主持人大赛>>有N位评委给选手评分，分数范围为0到100之间的浮点数
 *      选手的最终得分为：去掉最高分和最低分之后的N-2位评委的平均分
 *      训练目标：方法的设计
 */
public class CalTotalAvg {
    public static void main(String[] args) {
        double[] scores = inputData(6); //1)录入评委的评分
        double avg = calAvg(scores); //2)计算平均分
        System.out.println("平均分为:"+avg);
    }

    /** 录入N位评委的评分 */
    public static double[] inputData(int count){
        double[] scores = new double[count]; //评分数组
        Scanner scan = new Scanner(System.in);
        for(int i=0;i<scores.length;i++){
            System.out.println("请输入第"+(i+1)+"个评委的分数:");
            scores[i] = scan.nextDouble();
        }
        return scores;
    }

    /** 计算平均分 */
    public static double calAvg(double[] scores){
        double total = 0.0; //总分
        double max = scores[0]; //假设第1个元素为最高分
        double min = scores[0]; //假设第1个元素为最低分
        for(int i=0;i<scores.length;i++){
            if(scores[i]>max){ //找最高分
                max = scores[i];
            }
            if(scores[i]<min){ //找最低分
                min = scores[i];
            }
            total += scores[i]; //累加所有评分
        }
        //计算平均分--总分减掉最高分和最低分之后，再除以(评委数-2)
        double avg = (total-max-min)/(scores.length-2);
        return avg;
    }
}

```

```
}  
}
```

补充：

1. 明日单词：

- 1) **student**: 学生
- 2) **className**: 班级名称
- 3) **stuId**: 学号
- 4) **study**: 学习
- 5) **play**: 玩
- 6) **another**: 另一个
- 7) **car**: 小汽车
- 8) **brand**: 品牌
- 9) **color**: 颜色
- 10) **price**: 价格
- 11) **start**: 开始、启动
- 12) **run**: 跑
- 13) **stop**: 停止