Java方向编程题答案

day25

[编程题]23649-学分绩点

链接：https://www.nowcoder.com/questionTerminal/a0c09a7e0da04e728810a8aca7226b7b

【题目解析】

无

【解题思路】

该题目难度一般，主要处理好程序的输入，计算区间能够全覆盖即可。

【代码示例】

```
import java.util.Scanner;
public class Main{
    public static void main(String[] args){
        Scanner in=new Scanner(System.in);
        int n=in.nextInt();
        int gradePoint[]=new int[n];//每门课学分
        int coursePoint[]=new int[n];//每门课分数

        for(int i=0;i<n;i++){
            gradePoint[i]=in.nextInt();
        }
        for(int i=0;i<n;i++){
            coursePoint[i]=in.nextInt();
        }
        in.close();
        System.out.printf("%.2f",gradePointAverage(gradePoint,coursePoint));//格式输出

    }
    //根据实际成绩判断学分方法
    public static double pointOfEachCourse(int grade){
        double point = 0.0;
        if(grade>=90&&grade<=100){
            point=4.0;
        }else if(grade>=85&&grade<=89){
            point=3.7;
        }else if(grade>=82&&grade<=84){
            point=3.3;
        }else if(grade>=78&&grade<=81){
            point=3.0;
        }else if(grade>=75&&grade<=77){
            point=2.7;
        }else if(grade>=72&&grade<=74){
            point=2.3;
        }else if(grade>=68&&grade<=71){
            point=2.0;
        }else if(grade>=64&&grade<=67){
            point=1.5;
        }else if(grade>=60&&grade<=63){
            point=1.0;
        }else if(grade<60){
            point=0.0;
        }
        return (double)point;
    }
    //计算平均绩点的方法
    public static double gradePointAverage(int[] gradePoint,int[] coursePoint){
        double gradePointTotal=0;
        double gradeOfEachCourseTotal=0;
        for(int i=0;i<gradePoint.length;i++){
            gradePointTotal+=pointOfEachCourse(coursePoint[i])*gradePoint[i];
            gradeOfEachCourseTotal+=gradePoint[i];
        }
        return (double)gradePointTotal/gradeOfEachCourseTotal;
    }
}
```

[编程题]36919-人民币转换

链接：https://www.nowcoder.com/questionTerminal/00ffd656b9604d1998e966d555005a4b

【题目解析】

生活中开发票常见，将人民币数字表示转换为汉字表示

【解题思路】

- 第一步：解析字符串
- 第二步：整数部分：按照 亿，万，千百十个位处理
- 第三步：小数部分：按照 角，分处理

【示例代码】

```
import java.util.Scanner;
public class Main{

    //阿拉伯数字0-9对应的中文
    static String[] map = {"壹","贰","叁","肆","伍","陆","柒","捌","玖"};

    public static void main(String[] args){
         Scanner scan = new Scanner(System.in);
         while(scan.hasNext()){
             String number = scan.next();
             resolve(number);
         }
         scan.close();
    }


    public static void resolve(String str){
        String[] strArr = str.split("\\.");
        //整数部分
        int number = Integer.valueOf(str.split("\\.")[0]);

        StringBuffer res = new StringBuffer("人民币");
        int yi = (int)(number/100000000);
        if(yi!=0){
            res.append(resolveQian(yi)).append("亿");
            number = number-yi*100000000;
        }

        int wan = (int)(number/10000);
        if(wan!=0){
            res.append(resolveQian(wan)).append("万");
            number = number-wan*10000;
        }

         //处理千百十个位
        String beforePointString = resolveQian(number);
        if(beforePointString.length()>1){
             res.append(beforePointString).append("元");
        }

        //若有小数点，处理小数点后面位数
        //小数部分处理
        if(strArr.length>1){
             String afterPointStr = strArr[1];
             res.append(handleNumberAfterPoint(afterPointStr));
        }

        //在resolveQian() 方法里可能会返回  零xxx
        //但在最高为不能有"零"
        String resString = res.toString();

        if(resString.length()>4 && resString.charAt(3)=='零' && resString.charAt(4)!='元'){
            //最高位的零去掉
            resString = resString.substring(0,3)+resString.substring(4);
        }
        System.out.println(resString);

    }

    //处理4位数 千百十个位
    public static String resolveQian(double temp){

        StringBuffer resBuffer = new StringBuffer();

        //千位
        int qian = (int)(temp/1000);
        if(qian!=0){
            resBuffer.append(map[qian-1]).append("仟");

            temp = temp-qian*1000;
        }

        //百位
        int bai = (int)(temp/100);
        if(bai!=0){
            resBuffer.append(map[bai-1]).append("佰");
            temp = temp-bai*100;
        }
        //注意:零 只会添加在 百位和十位
        if(qian!=0 && bai==0){
            resBuffer.append("零");
        }

        //十位
        int shi = (int)(temp/10);
        if(shi!=0){
            if(shi!=1){
                resBuffer.append(map[shi-1]);
            }
            resBuffer.append("拾");
            temp = temp-shi*10;
        }

        //注意: 0
        if(bai!=0&&shi==0){
            resBuffer.append("零");
        }

        //个位
        int ge = (int)(temp%10);


        if(ge!=0){
            //5,0001 这种情况，千百十位均为0
            if(qian==0&&bai==0&&shi==0){
                resBuffer.append("零");
            }
            resBuffer.append(map[ge-1]);
        }
        String res = resBuffer.toString();
        return res;
    }

    //处理小数点后面的数
    public static String handleNumberAfterPoint(String str){
        String res = "";
        if(str.equals("00") ||str.equals("0")){
            res = "整";
        }else{
            if(str.charAt(0)!='0'){
                res += map[Integer.valueOf(str.charAt(0)+"")-1]+"角";
            }

            if(str.length()>1 && str.charAt(1)!='0'){
                res += map[Integer.valueOf(str.charAt(1)+"")-1]+"分";
            }
        }
        return res;
    }
}
```