

Java方向编程题答案

day42

[编程题]36897-计算日期到天数转换

<https://www.nowcoder.com/practice/769d45d455fe40b385ba32f97e7bcded?tpId=37& tqId=21296&rp=1&ru=/activity/oj&qu=/ta/huawei/question-ranking>

【题目解析】：该题目大意为：当用户输入今天的日期的时候，对应输出今天是今年的第几天。比如输入日期：2019 5 16 对应输出：136。

【解题思路】：我们的第一反应是：如果是5月，直接将前4个月的天数加起来，再加上5月的日期就好了。这里要注意，可不敢加5月这一个月天数。第二个需要注意的是：2月有可能是28天，有可能是29天。主要取决于是否为闰年。

【示例代码】：

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in) ;
        while(sc.hasNext()){
            int year = sc.nextInt() ;
            int month = sc.nextInt() ;
            int day = sc.nextInt() ;
            int Day = outDay(year, month, day) ;
            System.out.println(Day);
        }
        sc.close();
    }

    private static int outDay(int year, int month, int day) {
        //定义数组，这里2月的天数取28，假设是平年。
        int [] Day = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31} ;
        //对年进行添加判断以及求余
        if(year <= 0 || month <= 0 || month > 12 || day<= 0 || day > Day[month - 1])
            return -1;
        //如果是闰年，那么2月应该有29天。
        if(year % 4 == 0 && year % 100 != 0 || year % 400 == 0){
            Day[1] = 29 ;
        }

        int sum = 0 ;
        //这里要注意，如果是5月14日，5月这一个月不能算进去，算前4个月。
        for(int i = 0 ; i < month - 1 ; i++){
            sum += Day[i] ;
        }
        //最后sum需要加上当前月份的日期。
        return sum + day;
    }
}
```

```
}
```

[编程题]36853-字符串加解密

<https://www.nowcoder.com/practice/2aa32b378a024755a3f251e75cbf233a?tpId=37&&tqId=21252&rp=1&ru=/activity/oj&gru=/ta/huawei/question-ranking>

【题目解析】：童鞋们需要注意这个题型又是对字符串操作的考察，看似题目很长，实则就是“纸老虎”。我们来看一下：假如给定实例：abcdefg经过加密之后，就变为了BCDEFGH,字符a的后一个字符b对应的大写B.依次类推。解密：BCDEFGH--》刚刚的逆过程。就是abcdefg。

【解题思路】：我们读入用户输入的字符串，对字符串当中的每一个字符进行修改，随后添加到结果字符串当中。在这里我们使用了函数：public static String enCryption(String s){}; 在此函数内部我们做了这样的工作。对每个字符的加密，我们使用了函数：public static char encryption(char c){}; 在此函数内部，运用规则进行加密。相反解密的过程同上。分别使用函数：public static String deCryption(String s){}; 和public static char decryption(char c){};

【示例代码】：

```
import java.util.*;
public class Main {
    //对字符进行加密，加密原理：当前字符的后一个字符并且是大写
    //如果是数字则加1就好了，这里要注意如果是小写z,那么对应大写A.
    //大写Z对象小写a.
    public static char encryption(char c) {
        if(c >= 'a' && c < 'z')
            return (char)(c + 1 - 32);
        else if(c == 'z')
            return 'A';
        else if(c >= 'A' && c < 'Z')
            return (char)(c + 1 + 32);
        else if(c == 'Z')
            return 'a';
        else if(c >= '0' && c < '9')
            return (char)(c + 1);
        else if(c == '9')
            return '0';
        else
            return c;
    }
    //字符解密过程
    public static char decryption(char c) {
        if(c > 'a' && c <= 'z')
            return (char)(c - 1 - 32);
        else if(c == 'a')
            return 'Z';
        else if(c > 'A' && c <= 'Z')
            return (char)(c - 1 + 32);
        else if(c == 'A')
            return 'z';
        else if(c > '0' && c <= '9')
            return (char)(c - 1);
    }
}
```

```

        else if(c == '0')
            return '9';
        else
            return c;
    }
    //加密, 将输入的字符串中的每个字符进行加密。
    public static String enCryption(String s) {
        char[] cs = s.toCharArray();
        StringBuffer sb = new StringBuffer();
        for(int i = 0; i < cs.length; i++){
            sb.append(encryption(cs[i]));
        }
        return sb.toString();
    }
    //解密
    public static String deCryption(String s) {
        char[] cs = s.toCharArray();
        StringBuffer sb = new StringBuffer();
        for(int i = 0; i < cs.length; i++){
            sb.append(decryption(cs[i]));
        }
        return sb.toString();
    }

    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        while(sc.hasNext()){
            String s1 = sc.nextLine();//加过密的密码
            String s2 = sc.nextLine();//解密的密码
            System.out.println(enCryption(s1));
            System.out.println(deCryption(s2));
        }
        sc.close();
    }
}

```