

基础题目

第一题：需求实现

- 反转键盘录入的字符串。
- 代码实现，效果如图所示：

```
字符串123反转  
录入的字符串:字符串123反转  
反转的字符串:转反321串符字
```

- 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：

```
public class Test1 {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        String next = scanner.next();  
        System.out.println("录入的字符串:" + next);  
        String s = reverseStr(next);  
        System.out.println("反转的字符串:"+ s);  
    }  
  
    public static String reverseStr(String str){  
        String s = "";  
        char[] chars = str.toCharArray();  
        for (int i = chars.length - 1; i >= 0; i--) {  
            s +=chars[i] ;  
        }  
        return s;  
    }  
}
```

第二题：需求实现

- 键盘录入QQ号码，验证格式的正确性。
 - 必须是5—12位数字。
 - 0不能开头。
- 代码实现，效果如图所示：



```
237845546
这个QQ号码是否正确:true
```

- 参考答案：

```
public class Test2 {
    public static void main(String[] args) {
        //1.键盘输入一个qq号码字符串
        Scanner sc = new Scanner(System.in);
        String qq = sc.next();
        //2.调用checkQQ (String qq)方法内实现验证。
        boolean isOK = checkQQ(qq);
        //3.打印验证的结果
        System.out.println("这个QQ号码是否正确:" + isOK);
    }

    /*
     * 定义方法：checkQQ (String qq)方法内实现验证
     * 指定方法的名称：checkQQ
     * 指定方法的参数：String qq
     * 指定方法的返回值：boolean
     */
    public static boolean checkQQ(String qq){
        //1.验证字符串的长度5-12位之间；
        if (qq.length()<5 || qq.length()>12 ) {
            return false; //说明qq号码的长度不正确
        }
        //2.验证首位字符不能是字符0；只能是字符'1'--'9'
        if (qq.charAt(0) == '0') {
            return false;
        }
        //3.验证字符串中的每个字符都必须是数字字符'0'-'9'之间的字符
        for (int i = 0; i < qq.length(); i++) {
            char ch = qq.charAt(i);
            //判断字符是否在 数字字符'0'-'9'之间的字符
            if (ch<'0' || ch>'9') {
```

```

        return false;//说明qq号码中含有非数字字符
    }
}
//4.上述验证都通过了，说明qq号码是正确的
return true;
}
}

```

第三题：需求实现

- 字符串查找。
 - 键盘录入一个大字符串，再录入一个小字符串。
 - 统计小字符串在大字符串中出现的次数。

- 代码实现，效果如图所示：

```

请输入大字符串:woaiheima,heimabutongyubaima,wulunheima
请输入小字符串:heima
小字符串heima,在大字符串中共出现3次

```

- 参考答案：

```

public class Test3 {
    /*
     * 分析以下需求，并用代码实现
     * 1. 键盘录入一个大字符串,再录入一个小字符串
     * 2. 统计小字符串在大字符串中出现的次数
     * 3. 代码运行打印格式：
     *    请输入大字符串:woaiheima,heimabutongyubaima,wulunheimahaishibaima,zhaodaogongzuojiushihaoma
     *    请输入小字符串:heima
     *    控制台输出:小字符串heima,在大字符串woaiheima,heimabutongyubaima,wulunheimahaishibaima,zhaodaogongzuojiushihaoma中共出现3次
     */

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // 1. 键盘录入一个大字符串,再录入一个小字符串
        System.out.print("请输入大字符串:");
        String big = sc.nextLine();
        System.out.print("请输入小字符串:");
        String small = sc.nextLine();
        // 2. 统计小字符串在大字符串中出现的次数
    }
}

```

```

        int count = getCount(big, small);
        // 3.代码运行打印格式：
        System.out.println("小字符串" + small + ",在大字符串中共出现" + count +
"次");
    }

    /**
     * 方法功能：统计小字符串在大字符串中出现的次数
     * 参数：big 代表大字符串
     * 参数：small 代表小字符串
     * 返回值：小字符串在大字符串中出现的次数
     */
    public static int getCount(String big, String small) {
        int index = 0;
        int count = 0;
        /**
         * indexOf(String str, int fromIndex)
         * 该方法作用：从fromIndex位置开始查找，字符串str第一次出现的位置；若没找到，放回-1
         */
        while ((index = big.indexOf(small, index)) != -1) {
            index++;
            count++;
        }
        return count;
    }
}

```

第四题：需求实现

- 替换某字符串中的某字符串。
 - 键盘录入一个srcStr字符串，再录入一个delStr字符串。
 - 删除该srcStr字符串中的所有delStr字符串。
- 代码实现，部分效果如图所示：

```

请输入源字符串:java woaivava,i like jajavava i enjoy java
请输入要删除的字符串:ja
源字符串中总共包含:5 个 ja ,
删除ja后的字符串为: va woaiva,i like vava i enjoy va

```

- 参考答案：

```

public class Test4 {
    public static void main(String[] args) {

```

```

Scanner sc = new Scanner(System.in);
// 1. 键盘录入一个源字符串由字符串变量srcStr接收
System.out.print("请输入源字符串:");
String srcStr = sc.nextLine();
// 2. 键盘录入一个要删除的字符串由字符串变量delStr接收
System.out.print("请输入要删除的字符串:");
String delStr = sc.nextLine();
// 3. 删除该字srcStr字符串中的所有delStr字符串（最终的字符串中不能包含delStr），要求
打印删除后的结果以及删除了几个delStr字符串
printCount(srcStr, delStr);
}

//删除该字srcStr字符串中的所有delStr字符串（最终的字符串中不能包含delStr），要求打印删除
后的结果以及删除了几个delStr字符串
private static void printCount(String srcStr, String delStr) {
    //删除后的结果
    String resultStr = srcStr.replace(delStr, "");
    //删除了几个delStr字符串
    int count = (srcStr.length() - resultStr.length()) / delStr.length();

    System.out.println("源字符串中总共包含:" + count + " 个 " + delStr + " ,\r\n删
除"+ delStr +"后的字符串为: " + resultStr);
}
}

```

第五题：需求实现

- 生成一个随机100内小数，转换为保留两位小数的字符串，不考虑四舍五入的问题。
- 代码实现，效果如图所示：

```

随机数为：
86.21890326303406
转换为：
86.21

```

- 参考答案：

```

public class Test5 {
    public static void main(String[] args) {
        double random = Math.random()*100;
        System.out.println("随机数为:");
        System.out.println(random);
        String str = random+"";
        int index = str.indexOf(".");
        //System.out.println(index);
    }
}

```

```

        String substring = str.substring(0, index + 3);
        System.out.println("转换为:");
        System.out.println(substring);
    }
}

```

第六题：需求实现

- 筛选字符串。
 - 定义ArrayList集合，存入多个字符串。
 - 长度大于5的字符串，打印删除后的集合。
- 代码实现，效果如图所示：

```

源字符串：
[bca, dadfa, dddaaa, 你好啊, 我来啦, 你干嘛呢, 别跑啊]
删除后：
[bca, dadfa, 你好啊, 别跑啊]

```

- 参考答案：

```

public class Test6 {
    public static void main(String[] args) {
        ArrayList<String> list = getArrayList();
        System.out.println("源字符串:");
        System.out.println(list);
        //删除集合中长度大于5的元素
        delStrsFromList01(list);
        System.out.println("删除后:");
        System.out.println(list);
    }

    //返回存储多个字符串的ArrayList集合
    private static ArrayList<String> getArrayList() {
        ArrayList<String> list = new ArrayList<>();
        list.add("bca");
        list.add("dadfa");
        list.add("dddaaa");
        list.add("你好啊");
        list.add("我来啦, 你干嘛呢");
        list.add("别跑啊");
        return list;
    }
}

```

```

//删除list集合中所有长度>5的字符串
private static void delStrsFromList01(ArrayList<String> list) {
    //创建ArrayList集合对象
    ArrayList<String> newList = new ArrayList<String>();
    //遍历原集合对象
    for(int i = 0;i<list.size();i++) {
        //获取当前元素
        String str = list.get(i);
        //如果str的长度大于等于5
        if(str.length()>5) {
            newList.add(str);//将str添加到新集合中
        }
    }
    //遍历新集合
    for(Object str : newList) {
        //因为新集合中的元素就是要删除的元素
        list.remove(str);
    }
}
}

```

第七题：需求实现

- 判断回文字符串。如果一个字符串，从前向后读和从后向前读，都是一个字符串，称为回文串，比如mom，dad，noon。
- 代码实现，效果如图所示：

```

noon
回文数:true

```

- 参考答案：

```

public class Test7 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String next = scanner.next();
        boolean p = isP(next);
        System.out.println(
            "回文数:" + p
        );
    }
}

```

```

    public static boolean isP(String str) {
        int start = 0;
        int end = str.length() - 1;
        while (start < end) {
            if (str.charAt(start) != str.charAt(end)) {
                return false;
            }
            start++;
            end--;
        }
        return true;
    }
}

```

扩展题目

第八题：需求实现

- 模拟简单计算器，可以运算+，-，*，/，%。
 - 接收三个参数，一个整数，一个运算符，另一个整数。
 - 计算出运算结果。
 - 无法运算时，返回null。
- 代码实现，效果如图所示：

```

5
+
7
5+7=12

```

- 参考答案：

```

public class Test8 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int a = scanner.nextInt();
        String next = scanner.next();
        int b = scanner.nextInt();
        String count = count(a, next, b);
        System.out.println(a +next +b +"="+count);
    }
}

```



```

    public static String count(int a, String op , int b ){
        int r=0;
        if ("+".equals(op)){
            r = a+b;
        }else if ("-".equals(op)){
            r = a-b;
        }else if ("*".equals(op)){
            r = a*b;
        }else if ("/".equals(op)){
            r = a/b;
        }else if ("%".equals(op)){
            r = a%b;
        }else {
            return null;
        }

        return r+"";
    }
}

```

第九题：需求实现

- 校验密码是否合法。
 - 必须至少8个字符。
 - 必须至少2个大写字符。
 - 必须只有字母和数字。
- 代码实现，效果如图所示：

qweRY123 密码是否合法:true

- 参考答案：

```

public class Test9 {
    public static void main(String[] args) {
        String s = "qweRY123";
        System.out.println(s+" 密码是否合法:"+isV(s));
    }

    public static boolean isV(String pwd){
        if (pwd.length()<8){

```

```

        return false;
    }

    int countA = 0;
    char[] chars = pwd.toCharArray();
    for (int i = 0; i < chars.length; i++) {
        char ch = chars[i];
        // 2个 大写字符
        if (ch >= 'A' && ch <= 'Z'){
            countA++;
        }

        // 字母数字
        if ((ch < '0' || ch > '9') && (ch < 'A' || ch > 'Z') && (ch < 'a' || ch > 'z'))
    {

        return false;
    }
    }
    if (countA < 2) {
        return false;
    }
    return true;
}
}

```

第十题：需求实现

- 模拟用户登录。
 - 定义用户类，属性为用户名和密码。
 - 使用集合存储多个用户对象。
 - 录入用户和密码，对比用户信息，匹配成功登录成功，否则登录失败。
 - 登录失败时，当用户名错误，提示没有该用户。
 - 登录失败时，当密码错误时，提示密码有误。
- 代码实现，效果如图所示：

```

jack-1234
rose-5678
tom-0000
请输入用户名
rose
请输入密码
8888
登录结果:密码错误

```

■ 参考答案:

```
public class Test10 {
    static ArrayList<User> list = new ArrayList<>();
    static {
        list.add(new User("jack", "1234"));
        list.add(new User("rose", "5678"));
        list.add(new User("tom", "0000"));
        for (int i = 0; i < list.size(); i++) {
            list.get(i).show();
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入用户名");
        String username = sc.nextLine();
        System.out.println("请输入密码");
        String password = sc.nextLine();
        User u = new User(username,password);
        String login = login(u);
        System.out.println("登录结果:"+login);
    }

    public static String login(User user){
        String msg = "";
        String n = user.getUsername();
        String p = user.getPwd();
        for (int i = 0; i < list.size(); i++) {
            User u = list.get(i);
            String name = u.getUsername();
            String pwd = u.getPwd();
            if(name.equals(n)) {
                if (pwd.equals(p)) {
                    return "登录成功";
                }else {
                    // 密码错误
                    return "密码错误";
                }
            }else{
                msg = "用户名不存在"; //用户名不匹配
                continue;
            }
        }
    }
}
```

```

        }
        return msg ;
    }
}

class User{
    private String username;
    private String pwd;

    public User() {
    }

    public User(String username, String pwd) {
        this.username = username;
        this.pwd = pwd;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPwd() {
        return pwd;
    }

    public void setPwd(String pwd) {
        this.pwd = pwd;
    }

    public void show() {
        System.out.println(username+"-"+pwd);
    }
}

```