

知识点

String、StringBuilder

题目1（加强训练）

为了提高大家的上课听课效率，防止同学们上课走神，下课时王老师找到学校的技术老师陈老师，希望陈老师帮忙写一套随机点名器程序，辅助教学。因为着急回班级授课，王老师交代完需求后匆忙报了一串班级人员姓名就回去了，名单如下：

"令狐冲, 风清扬,左冷禅,岳不群,苗人凤";

请使用所学知识设计程序帮助陈老师实现随机点名器功能，要求实现如下效果：

本班人员名单如下：

令狐冲 风清扬

左冷禅 岳不群

苗人凤

本次被选中的幸运同学是：岳不群

训练目标

能够使用字符串的常用方法实现字符串的分割，遍历。

训练提示

- 1、老王丢下的人员名单是个很长的字符串，考虑通过逗号分割将大串转成数组操作比较方便。
- 2、班级人员名单的输出其实就是对分割后的字符串数组进行遍历，每输出两个就换一行。
- 3、随机点名其实就是利用随机数生成一个数组的随机索引，然后根据索引获取数组元素打印即可。

参考方案

使用随机数结合字符串相关API方法完成

操作步骤

- 1、定义一个字符串,用来维护点名表人员名单
- 2、对名单字符串进行拆分,通过逗号分割names,得到所有人名的数组
- 3、遍历输出人名数组,没输出两个人名就换一行
 - 3.1、当i对2取余为1时,输出元素后换行,相当于每输出2个人名就换行
- 4、输出一个换行
- 5、通过随机数,生成一个随机索引

6、输出随机索引位置对应的数组元素,也就是被点名的人姓名

参考答案

```
import java.util.Random;

public class Test01 {
    public static void main(String[] args) {
        //1. 定义一个字符串,用来维护点名表人员名单
        String names = "令狐冲, 风清扬,左冷禅,岳不群,苗人凤";

        //2. 对名单字符串进行拆分,通过逗号分割names,得到所有人名的数组
        String[] nameArray = names.split(",");

        //3. 遍历输出人名数组,没输出两个人名就换一行
        System.out.println("本班人员名单如下: ");
        for (int i = 0; i < nameArray.length; i++) {
            //3.1 当i对2取余为1时,输出元素后换行,相当于每输出2个人名就换行
            if (i % 2 == 1) {
                System.out.println(nameArray[i]);
            } else {
                System.out.print(nameArray[i] + "\t");
            }
        }
        //4. 输出一个换行
        System.out.println();

        //5. 通过随机数,生成一个随机索引
        Random r = new Random();
        int index = r.nextInt(nameArray.length);

        //6. 输出随机索引位置对应的数组元素,也就是被点名的人姓名
        System.out.println("本次被选中的幸运同学是: "+nameArray[index]);
    }
}
```

视频讲解

另附avi文件提供。

题目2 (综合案例)

随着视频直播的兴起,弹幕评论是当下比较热门的技术。在实际业务开发中,通常会对用户的弹幕评论信息进行处理,将一些带有攻击性或者敏感的字眼和谐掉,技术上称之为“敏感词过滤”。

请设计程序模拟敏感词过滤:使用键盘录入模拟弹幕评论,将录入的数据进行处理,要求不能出现: sb, tmd, 傻蛋等字眼(这里字母不区分大小写匹配),如果出现了,则使用**代替,将过滤后的内容打印到控制台,效果如下:

相信你的评论,一定可以一针见血:

Sb,你*Tmd*个*sha*蛋,再见~

******,你******个******,再见~

Process finished with exit code 0

相信你的评论,一定可以一针见血:

sb,你个傻蛋,*tmd*

******,你个******,******

Process finished with exit code 0

相信你的评论,一定可以一针见血:

SB,你*TMD*个傻蛋,再见~

******,你******个******,再见~

Process finished with exit code 0

训练目标

能够使用String类的方法完成敏感词过滤。

训练提示

- 1、敏感词过滤,实际就是拿******替换敏感的字眼,替换方法就是replace
- 2、从结果上来看,这里要考虑到敏感词的字母大小写以及汉字的拼音组合情况,可以考虑将所有情况都整理到数组容器中管理
- 3、应该使用上一次替换后的结果继续下一个敏感词的替换,这样才能保证最终的结果是所有敏感词都替换后的值

参考方案

使用String类的replace方法完成

操作步骤

- 1、定义String类型变量s,接收键盘录入的评论信息
- 2、定义一个数组,用来表示敏感词字典
- 3、遍历敏感字典数组
 - 3.1、拿到单个敏感词
 - 3.2、判断录入的评论信息是否包含当前的敏感词,如果包含则替换为******
 - 3.3、调用s对象的replace方法,替换敏感词,将替换后的结果重新赋值给s
- 4、输出过滤后的结果s

参考答案

```
import java.util.Scanner;

public class Test02 {
    public static void main(String[] args) {
        //1. 定义String类型变量s,接收键盘录入的评论信息
        Scanner sc = new Scanner(System.in);
        System.out.println("相信你的评论,一定可以一针见血:");
        String s = sc.nextLine();

        //2. 定义一个数组,用来表示敏感词字典
        String[] sensitivewords =
{"sb","Sb","sB","SB","tmd","Tmd","tMd","tmD","Tmd","tMD","Tmd","TMD","傻蛋","sha蛋","傻dan"};
    }
```

```

//3. 遍历敏感字典
for (int i = 0; i < sensitivewords.length; i++) {
    //3.1 拿到单个敏感词
    String sensitiveword = sensitivewords[i];
    //3.2 判断录入的评论信息是否包含当前的敏感词，如果包含则替换为**
    if (s.contains(sensitiveword)){
        //3.3 调用s对象的replace方法，替换敏感词，将替换后的结果重新赋值给s
        s = s.replace(sensitiveword, "**");
    }
}

//4. 输出过滤后的结果s
System.out.println(s);
}
}

```

视频讲解

另附avi格式视频。

题目3（综合扩展）

在我们进行网站登录注册时，通常都会有一个输入验证码的选项。验证码的组成通常包含：大写字母，小写字母和数字，验证码的长度通常为4个字符长度，不同字符的搭配完全随机。

请使用所学知识，定义一个方法，专门用于生成随机验证码，并在主方法中完成调用，模拟用户验证码输入校验过程，当用户输入的验证码与系统生成验证码匹配时，给出验证通过提示，否则，重新生成验证码，要求用户重新输入，再重复以上匹配逻辑。实现如下效果：

当前验证码：EON7

abcd

当前验证码：3qQ5

abcd

当前验证码：vHq5

45gd

当前验证码：BWyE

BwYe

验证通过...

Process finished with exit code 0

训练目标

能够使用StringBuilder完成字符串拼接，使用字符串忽略大小写比较方法完成验证码的匹配。

训练提示

- 1、验证码中每个字符的取值范围确定好了，可以考虑将所有字符汇总成一个大的字符串。
- 2、要生成验证码，其实就是从大的字符串中随机取4个索引，然后将每个索引位置的字符拿到拼接成一个完整的字符串
- 3、从最终效果上来看，验证码的匹配应该是忽略大小写的，且匹配错误后会重新生成一个新的验证码。

参考方案

使用Random，StringBuilder，String完成。

操作步骤

- 1、定义一个方法，专门用来生成验证码
 - 1.1、定义一个字符串表示大写字母所有取值范围
 - 1.2、将大写字母对应的字符串调用toLowerCase()得到所有小写字母取值范围
 - 1.3、定义一个字符串表示所有数字取值范围
 - 1.4、定义一个StringBuilder变量 sb，用于将所有字符拼接到一起
 - 1.5、遍历小写字母字符串，将每一个小写字母字符拼接 to StringBuilder 中
 - 1.6、遍历数字字符串，将每一个数字字符拼接 to StringBuilder 中
 - 1.7、重新定义一个StringBuilder变量 sb1,用来拼接验证码
 - 1.8、循环四次,每次从sb中取一个随机字符
 - 1.9、将sb1转为字符串并返回
- 2、调用获取验证码的方法,将得到的验证码输出到控制台
- 3、通过键盘录入获取用户输入的验证码
- 4、比较用户输入的和键盘录入的验证码是否匹配，这里使用忽略大小写比较
 - 4.1、匹配正确，给出提示信息，结束循环

参考答案

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

public class Test03 {
    public static void main(String[] args) {
        //2、调用获取验证码的方法,将得到的验证码输出到控制台
        while (true) {
            String checkCode = getCheckCode();
            System.out.println("当前验证码: " + checkCode);
            //3、通过键盘录入获取用户输入的验证码
```

```

Scanner sc = new Scanner(System.in);
String userInput = sc.next();

//4、比较用户输入的和键盘录入的验证码是否匹配，这里使用忽略大小写比较
if (checkCode.equalsIgnoreCase(userInput)) {
    //4.1、匹配正确，给出提示信息，结束循环
    System.out.println("验证通过...");
    break;
}
}
}

//1、定义一个方法，专门用来生成验证码
public static String getCheckCode() {
    //1.1 定义一个字符串表示大写字母所有取值范围
    String upCode = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    //1.2 将大写字母对应的字符串调用toLowerCase()得到所有小写字母取值范围
    String lowerCode = upCode.toLowerCase();
    //1.3 定义一个字符串表示所有数字取值范围
    String numberCode = "0123456789";
    //1.4 定义一个StringBuilder变量 sb，用于将所有字符拼接到一起
    StringBuilder sb = new StringBuilder(upCode);
    //1.5 遍历小写字母字符串，将每一个小写字母字符拼接接到StringBuilder中
    for (int i = 0; i < lowerCode.length(); i++) {
        sb.append(lowerCode.charAt(i));
    }
    //1.6 遍历数字字符串，将每一个数字字符拼接接到StringBuilder中
    for (int i = 0; i < numberCode.length(); i++) {
        sb.append(numberCode.charAt(i));
    }
    //1.7 重新定义一个StringBuilder变量 sb1,用来拼接验证码
    StringBuilder sb1 = new StringBuilder();
    //1.8 循环四次，每次从sb中取一个随机字符
    Random r = new Random();
    for (int i = 0; i < 4; i++) {
        //使用Random生成随机索引
        int index = r.nextInt(sb.length());
        //获取随机索引位置对应的字符
        char ch = sb.charAt(index);
        //将获取到的字符拼接接到sb1中
        sb1.append(ch);
    }
    //1.9 将sb1转为字符串并返回
    return sb1.toString();
}
}

```

视频讲解

另附avi文件提供。