**《面向对象程序设计》课程实验报告**

**实验完成日期**： - -

|  |  |  |  |
| --- | --- | --- | --- |
| **课程名称**：**面向对象程序设计** | | **专业：**软件工程 | **教学班级：** |
| **姓名**：贠若轩 | | **学号**：2410250062 | **指导教师：** |
| **实验名称**：**实验5 Java API应用** | | | **实验学时：6学时** |
| **实验类型** | ☑验证性 □演示性 ☑设计性 □综合性 | | |
| **实验重点** | String、StringBuffer；System、Runtime；Math、Random；日期时间类、包装类、迭代器、Collection接口、List接口、Set接口；Map接口；容器元素的排序、查找和遍历 | | |
| **实验难点** | String类的常见操作、日期时间类、集合的查找和遍历方式 | | |
| **实验环境** | jdk8.0以上版本，Editplus、Eclipse、IDEA等集成开发环境均可 | | |

**一、实验目的**

1、掌握String、StringBuffer和StringBuilder类的使用；

2、掌握System和Runtime类的使用；

3、掌握Math和Random类的使用；

4、掌握日期时间类的使用；

5、掌握包装类的使用；

6、了解正则表达式的使用；

7、掌握容器类的层次结构；

8、掌握Collection接口和List接口的常用方法；

9、掌握Iterator接口的使用方式；

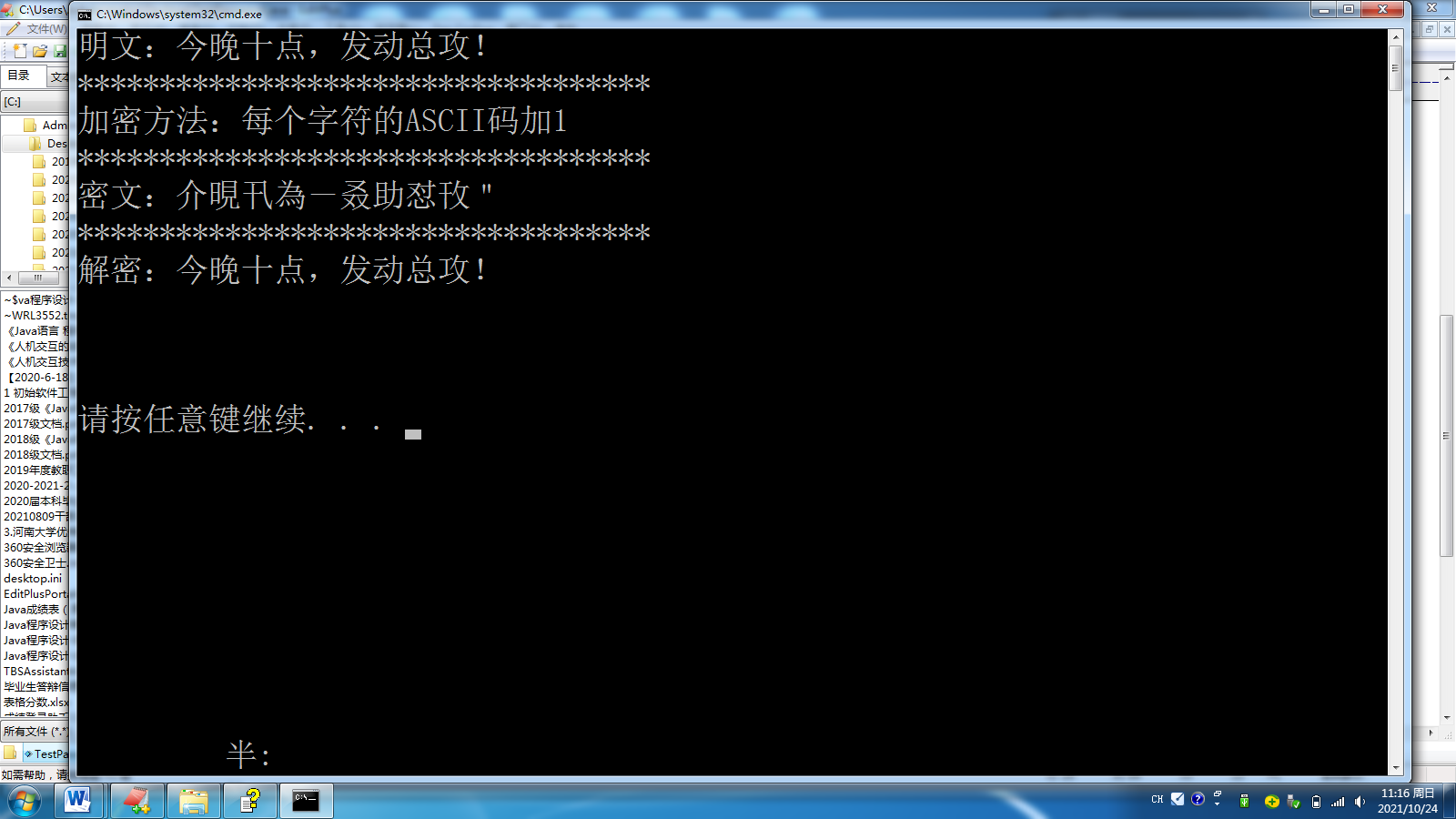
10、掌握Set接口和hashCode的使用方式；

11、掌握Map接口的使用方式。

**二、实验内容**

1、实验题目：字符串加密

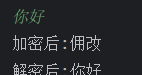
键盘输入一个原始字符串作为明文，然后使用加密方法加密，再对加密字符串进行解密。样例如下图，加密方法自定，完成其功能并测试。



**源代码：**

package qimo.shiyan5;  
  
import java.util.Scanner;  
  
public class StringCipher {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 String plaintext = scanner.nextLine();  
 scanner.close();  
  
 String encrypted = *encrypt*(plaintext);  
 String decrypted = *decrypt*(encrypted);  
  
 System.*out*.println("加密后:" + encrypted);  
 System.*out*.println("解密后:" + decrypted);  
 }  
  
 static String encrypt(String plaintext) {  
 StringBuilder encrypted = new StringBuilder();  
 for (char c : plaintext.toCharArray()) {  
 encrypted.append((char) (c + 3));  
 }  
 return encrypted.toString();  
 }  
  
 static String decrypt(String encrypted) {  
 StringBuilder decrypted = new StringBuilder();  
 for (char c : encrypted.toCharArray()) {  
 decrypted.append((char) (c - 3));  
 }  
 return decrypted.toString();  
 }  
}

**列出测试数据和实验结果截图：**



2、实验题目：模拟用户注册

编写一个程序，模拟用户注册和登录。程序要求如下：

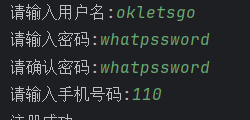
（1）输入用户名、密码、确认密码、手机号码，密码和确认密码一致则提示“注册成功”，并进入登录环节；

（2）密码和确认密码不一致，则提示“密码不一致，请重新注册”，重新进入注册环节；

（3）登录环节按照提示输入用户名和密码，密码正确（不区分大小写）则提示“登录成功”并打开Windows计算器，不正确则提示“密码错误，请重新输入！”，总共有3次登录机会，超过3次，则提示“登录失败，无法再继续登录”。

**源代码：**package qimo.shiyan5;  
  
import java.util.Scanner;  
import java.io.IOException;  
  
public class UserRegistration {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 String username = "", password = "", phone = "";  
 boolean registered = false;  
  
 while (!registered) {  
 System.*out*.print("请输入用户名:");  
 username = scanner.nextLine().trim();  
 System.*out*.print("请输入密码:");  
 password = scanner.nextLine();  
 System.*out*.print("请确认密码:");  
 String confirmPassword = scanner.nextLine();  
 System.*out*.print("请输入手机号码:");  
 phone = scanner.nextLine().trim();  
  
 if (password.equals(confirmPassword) && !username.isEmpty() && !phone.isEmpty()) {  
 System.*out*.println("注册成功");  
 registered = true;  
 } else {  
 System.*out*.println("密码不一致或信息不完整，请重新注册");  
 }  
 }  
  
 int loginAttempts = 0;  
 boolean loggedIn = false;  
  
 while (loginAttempts < 3 && !loggedIn) {  
 System.*out*.print("请输入用户名:");  
 String loginUsername = scanner.nextLine();  
 System.*out*.print("请输入密码:");  
 String loginPassword = scanner.nextLine();  
  
 if (loginUsername.equals(username) && loginPassword.equals(password)) {  
 System.*out*.println("登录成功");  
 loggedIn = true;  
 try {  
 if (System.*getProperty*("os.name").toLowerCase().contains("win")) {  
 Runtime.*getRuntime*().exec("calc");  
 } else {  
 System.*out*.println("计算器功能仅支持Windows系统");  
 }  
 } catch (IOException e) {  
 System.*out*.println("无法打开计算器: " + e.getMessage());  
 }  
 } else {  
 loginAttempts++;  
 System.*out*.println("密码错误，请重新输入！");  
 if (loginAttempts == 3) {  
 System.*out*.println("登录失败，无法再继续登录");  
 }  
 }  
 }  
 scanner.close();  
 }  
}

**列出测试数据和实验结果截图：**



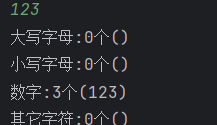
3、实验题目：统计字符个数

从键盘输入一个字符串，分别统计该字符串中所有大写字母、小写字母、数字、其它字符的个数，并分类输出这些字符和统计结果。（提示：不考虑字符内容，例如：Hello123World，大写2个，小写8个，数字3个。）

**源代码：**

package qimo.shiyan5;  
  
import java.util.Scanner;  
  
public class CharacterCounter {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 String input = scanner.nextLine();  
 scanner.close();  
  
 StringBuilder upper = new StringBuilder();  
 StringBuilder lower = new StringBuilder();  
 StringBuilder digit = new StringBuilder();  
 StringBuilder other = new StringBuilder();  
  
 for (char c : input.toCharArray()) {  
 if (Character.*isUpperCase*(c)) {  
 upper.append(c);  
 } else if (Character.*isLowerCase*(c)) {  
 lower.append(c);  
 } else if (Character.*isDigit*(c)) {  
 digit.append(c);  
 } else {  
 other.append(c);  
 }  
 }  
  
 System.*out*.println("大写字母:" + upper.length() + "个(" + upper + ")");  
 System.*out*.println("小写字母:" + lower.length() + "个(" + lower + ")");  
 System.*out*.println("数字:" + digit.length() + "个(" + digit + ")");  
 System.*out*.println("其它字符:" + other.length() + "个(" + other + ")");  
 }  
}

**列出测试数据和实验结果截图：**



4、实验题目：字符串缓冲区练习

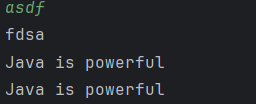
（1）使用StringBuffer类对键盘输入的字符串进行反转。

（2）使用String和StringBuffer类分别对数组进行字符串拼接，使其变成一个字符串。

**源代码：**

package qimo.shiyan5;  
  
import java.util.Scanner;  
  
public class StringBufferPractice {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 // 第一部分：字符串反转  
 String input = scanner.nextLine();  
 String reversed = new StringBuffer(input).reverse().toString();  
 System.*out*.println(reversed);  
  
 // 第二部分：字符串拼接  
 String[] arr = {"Java", "is", "powerful"};  
  
 // 使用 String  
 String resultStr = "";  
 for (String s : arr) {  
 resultStr += s + " ";  
 }  
 System.*out*.println(resultStr);  
  
 // 使用 StringBuffer  
 StringBuffer resultBuf = new StringBuffer();  
 for (String s : arr) {  
 resultBuf.append(s).append(" ");  
 }  
 System.*out*.println(resultBuf.toString());  
  
 scanner.close();  
 }  
}

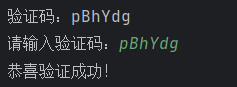
**列出测试数据和实验结果截图：**



5、实验题目：生成验证码

使用Random类创建一个6位的验证码，其中包含数字、字母的组合，并通过键盘输入该验证码，验证通过（不区分大小写）时提示“恭喜验证成功！”，否则提示“验证失败！”。

**源代码：**package qimo.shiyan5;  
  
import java.util.Random;  
import java.util.Scanner;  
  
public class VerificationCode {  
 public static void main(String[] args) {  
 String codeChars = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";  
 Random random = new Random();  
 StringBuilder code = new StringBuilder();  
  
 for (int i = 0; i < 6; i++) {  
 int index = random.nextInt(codeChars.length());  
 code.append(codeChars.charAt(index));  
 }  
  
 System.*out*.println("验证码：" + code);  
 System.*out*.print("请输入验证码：");  
 Scanner scanner = new Scanner(System.*in*);  
 String input = scanner.nextLine();  
 scanner.close();  
  
 if (input.equalsIgnoreCase(code.toString())) {  
 System.*out*.println("恭喜验证成功！");  
 } else {  
 System.*out*.println("验证失败！");  
 }  
 }  
}

**列出测试数据和实验结果截图：**

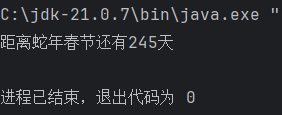
6、实验题目：春节倒计时

根据所学知识，计算明年（蛇年）春节的倒计时天数并输出：“距离蛇年春节还有\*\*\*天”。

**源代码：**

package qimo.shiyan5;  
  
import java.time.LocalDate;  
import java.time.temporal.ChronoUnit;  
  
public class SpringFestivalCountdown {  
 public static void main(String[] args) {  
 LocalDate today = LocalDate.*now*();  
 int currentYear = today.getYear();  
 LocalDate springFestival = LocalDate.*of*(currentYear + 1, 2, 10); // 2026年2月10日是蛇年春节  
  
 if (today.isAfter(springFestival)) {  
 springFestival = springFestival.plusYears(1);  
 }  
  
 long days = ChronoUnit.*DAYS*.between(today, springFestival);  
 System.*out*.println("距离蛇年春节还有" + days + "天");  
 }  
}

**列出测试数据和实验结果截图：**



7、实验题目：二月天

二月是一个有趣的月份，平年的二月有28天，闰年的二月有29天。编写一个程序，从键盘输入年份，根据输入的年份计算这一年的二月有多少天。

**源代码：**

package qimo.shiyan5;  
  
import java.util.Scanner;  
  
public class FebruaryDays {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 int year = scanner.nextInt();  
 scanner.close();  
  
 boolean isLeapYear = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);  
 int days = isLeapYear ? 29 : 28;  
  
 System.*out*.println(year + "年的二月有" + days + "天");  
 }  
}

**列出测试数据和实验结果截图：**



8、实验题目：正则表达式。（选做）

“中华人民共和国成立于1949年10月1日”，利用正则表达式提取出其中的数字。

**源代码：**package qimo.shiyan5;  
  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class RegexExtractNumbers {  
 public static void main(String[] args) {  
 String text = "中华人民共和国成立于1949年10月1日";  
 Pattern pattern = Pattern.*compile*("\\d+");  
 Matcher matcher = pattern.matcher(text);  
  
 while (matcher.find()) {  
 System.*out*.println(matcher.group());  
 }  
 }  
}

**列出测试数据和实验结果截图：**

9、将下列数据：“hello”、123、6.9、“hello”、“”、“Hello”、StringBuffer s=new StringBuffer(“hello”)中的s，添加到一个ArrayList对象中。。

• 将ArrayList中的所有元素打印输出。

• 查找元素“hello”。

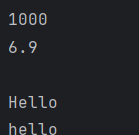
• 删除指定的元素“hello”。

• 将元素123替换为1000。

**源代码：**

package qimo.shiyan5;  
  
import java.util.ArrayList;  
  
public class ArrayListPractice {  
 public static void main(String[] args) {  
 ArrayList<Object> list = new ArrayList<>();  
 list.add("hello");  
 list.add(123);  
 list.add(6.9);  
 list.add("hello");  
 list.add("");  
 list.add("Hello");  
 StringBuffer s = new StringBuffer("hello");  
 list.add(s);  
  
 // 打印所有元素  
 System.*out*.println("初始ArrayList：");  
 for (Object element : list) {  
 System.*out*.println(element);  
 }  
  
 // 查找元素"hello"  
 System.*out*.println("\n元素\"hello\"的索引位置：");  
 for (int i = 0; i < list.size(); i++) {  
 if ("hello".equals(list.get(i))) {  
 System.*out*.println("索引 " + i);  
 }  
 }  
  
 // 删除所有"hello"元素  
 list.removeIf(element -> "hello".equals(element));  
 System.*out*.println("\n删除所有\"hello\"后的ArrayList：");  
 for (Object element : list) {  
 System.*out*.println(element);  
 }  
  
 // 替换元素123为1000  
 for (int i = 0; i < list.size(); i++) {  
 if (list.get(i).equals(123)) {  
 list.set(i, 1000);  
 }  
 }  
 System.*out*.println("\n替换123为1000后的ArrayList：");  
 for (Object element : list) {  
 System.*out*.println(element);  
 }  
 }  
}

**列出测试数据和实验结果截图：**



10、使用ArrayList集合，向集合中添加10个整数，并使用Iterator遍历该集合，并查找键盘输入的元素。提示：

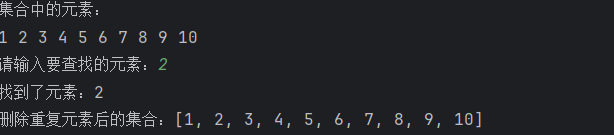
• 使用add()方法将元素添加到ArrayList集合中。

• 调用集合的iterator()方法获得Iterator对象，并调用Iterator的hasNext()和next()方法，迭代出集合中的所有元素，完成查找功能，并将重复的元素删除。

**源代码：**

package qimo.shiyan5;  
  
import java.util.ArrayList;  
import java.util.Iterator;  
import java.util.Scanner;  
  
public class ArrayListIterator {  
 public static void main(String[] args) {  
 ArrayList<Integer> list = new ArrayList<>();  
 for (int i = 1; i <= 10; i++) {  
 list.add(i);  
 }  
  
 // 使用Iterator遍历并打印元素  
 System.*out*.println("集合中的元素：");  
 Iterator<Integer> iterator = list.iterator();  
 while (iterator.hasNext()) {  
 System.*out*.print(iterator.next() + " ");  
 }  
 System.*out*.println();  
  
 // 查找键盘输入的元素  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("请输入要查找的元素：");  
 int target = scanner.nextInt();  
 scanner.close();  
  
 // 使用Iterator查找并删除重复元素  
 boolean found = false;  
 iterator = list.iterator();  
 while (iterator.hasNext()) {  
 int element = iterator.next();  
 if (element == target) {  
 if (found) {  
 iterator.remove(); // 删除重复元素  
 } else {  
 found = true;  
 }  
 }  
 }  
  
 // 输出查找结果  
 if (found) {  
 System.*out*.println("找到了元素：" + target);  
 System.*out*.println("删除重复元素后的集合：" + list);  
 } else {  
 System.*out*.println("未找到元素：" + target);  
 }  
 }  
}

**列出测试数据和实验结果截图：**



11、分别利用Arraylist和Set随机生成十个不重复的随机整数，随机整数范围为350到450。

**源代码：**

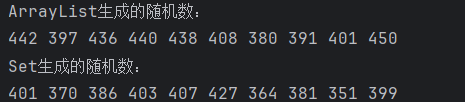
**列出测试数据和实验结果截图：**

12、集合中不容许有重复的对象，对于多个重复对象只能添加一次。例如在HashSet集合中添加三个Person对象，把姓名相同的人当做同一个人，虽然可以添加多次但集合里只保留一个，但是这对类的设计是有要求的，假设Person类中只包含name和age属性，则需要重写hashCode()方法和equals()方法，如果两个对象的name相同，则hashCode()方法的返回值相同，equals（）方法返回true。

**源代码：**

package qimo.shiyan5;  
  
import java.util.ArrayList;  
import java.util.HashSet;  
import java.util.Random;  
import java.util.Set;  
  
public class RandomNumberGenerator {  
 public static void main(String[] args) {  
 // 使用ArrayList生成  
 ArrayList<Integer> arrayList = new ArrayList<>();  
 Random random = new Random();  
 while (arrayList.size() < 10) {  
 int num = random.nextInt(101) + 350;  
 if (!arrayList.contains(num)) {  
 arrayList.add(num);  
 }  
 }  
 System.*out*.println("ArrayList生成的随机数：");  
 for (int num : arrayList) {  
 System.*out*.print(num + " ");  
 }  
 System.*out*.println();  
  
 // 使用Set生成  
 Set<Integer> set = new HashSet<>();  
 while (set.size() < 10) {  
 set.add(random.nextInt(101) + 350);  
 }  
 System.*out*.println("Set生成的随机数：");  
 for (int num : set) {  
 System.*out*.print(num + " ");  
 }  
 }  
}

**列出测试数据和实验结果截图：**



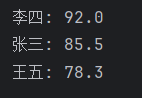
13、编写程序将一组学生对象的姓名和成绩存入到一个树集（TreeSet）中，完成以下要求：

• 使得按照成绩自动降序排列，并输出排序的结果。

**源代码：**

package qimo.shiyan5;  
  
import java.util.\*;  
  
class Student implements Comparable<Student> {  
 private String name;  
 private double score;  
  
 public Student(String name, double score) {  
 this.name = name;  
 this.score = score;  
 }  
  
 public double getScore() {  
 return score;  
 }  
  
 @Override  
 public int compareTo(Student other) {  
 return Double.*compare*(other.score, this.score);  
 }  
  
 @Override  
 public String toString() {  
 return name + ": " + score;  
 }  
}  
  
public class TreeSetStudentSort {  
 public static void main(String[] args) {  
 TreeSet<Student> students = new TreeSet<>();  
 students.add(new Student("张三", 85.5));  
 students.add(new Student("李四", 92.0));  
 students.add(new Student("王五", 78.3));  
 students.add(new Student("赵六", 92.0));  
  
 for (Student student : students) {  
 System.*out*.println(student);  
 }  
 }  
}

**列出测试数据和实验结果截图：**



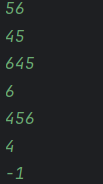
14、编写一个程序，读取个数不定的整数，然后查找其中出现频率最高的数字。要求通过键盘输入数据，当输入为0时，表示结束输入。如： 如果输入的数据是2 3 40 3 54 -3 3 3 2 0，那么数字3的出现频率是最高的。如果出现频率最高的数字不是一个而是多个，则应该将它们全部输出。例如当数据是9 30 3 9 3 2 4时，3和9都出现了两次，3和9都应该输出。

提示：可以利用集合的元素不能重复这一特性。

**源代码：**

package qimo.shiyan5;  
  
import java.util.\*;  
  
public class MostFrequentNumber {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 Map<Integer, Integer> countMap = new HashMap<>();  
  
 int num;  
 while ((num = scanner.nextInt()) != 0) {  
 countMap.put(num, countMap.getOrDefault(num, 0) + 1);  
 }  
 scanner.close();  
  
 if (countMap.isEmpty()) {  
 System.*out*.println("没有输入有效数字");  
 return;  
 }  
  
 int maxCount = Collections.*max*(countMap.values());  
 List<Integer> mostFrequent = new ArrayList<>();  
  
 for (Map.Entry<Integer, Integer> entry : countMap.entrySet()) {  
 if (entry.getValue() == maxCount) {  
 mostFrequent.add(entry.getKey());  
 }  
 }  
  
 for (int number : mostFrequent) {  
 System.*out*.print(number + " ");  
 }  
 }  
}

**列出测试数据和实验结果截图：**

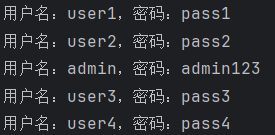


15、选择合适的Map集合保存5个用户的用户名和密码，然后将这些键值对打印出来。

**源代码：**

package qimo.shiyan5;  
  
import java.util.HashMap;  
import java.util.Map;  
  
public class UserLogin {  
 public static void main(String[] args) {  
 Map<String, String> userMap = new HashMap<>();  
 userMap.put("admin", "admin123");  
 userMap.put("user1", "pass1");  
 userMap.put("user2", "pass2");  
 userMap.put("user3", "pass3");  
 userMap.put("user4", "pass4");  
  
 for (Map.Entry<String, String> entry : userMap.entrySet()) {  
 System.*out*.println("用户名：" + entry.getKey() + "，密码：" + entry.getValue());  
 }  
 }  
}

**列出测试数据和实验结果截图：**



16、（选做）统计字符串中每个单词出现的次数，使用HashMap来实现。例如：“Today, We have a class of java, as we kown, java is an object oriented programming language, and java is fun! wish you enjoy it!”，统计结果存储成以下形式：

a-->1

an-->1

and-->1

as-->1……

is-->2

提示：使用String.split(("[ \n\t\r.,;:!?()]")方法进行分词。

**源代码：**

**列出测试数据和实验结果截图：**

17、（选做）500个人围成一个圈，从1开始报数，数到3的倍数的人离开圈子，循环往复，直到最后圈子只剩下一人为止，求剩下的人原来在圈子的位置。

提示：可以使用集合（ArrayList）或队列（Deque）实现。

**源代码：**

**列出测试数据和实验结果截图：**

**三、实验总结**

**对实验结果进行分析，列出错误及解决办法，回答问题，总结实验的心得体会，以及提出改进意见。**

实验中出现了未正确实现接口方法、输入验证不完善、计算器兼容性问题等错误，通过补全接口实现、增加异常处理和系统判断得以解决；体会到面向对象编程中抽象类与接口的差异及多态的灵活性，建议在后续开发中注重代码健壮性和用户交互友好性。

**注：**

**1、实验报告完成后，修改文件名为：实验5-实验报告。**

**2、将java源程序文件按照题号命名，如S5\_1.java、S5\_2.java、S5\_3.java，…，并将相关源代码和实验报告文件，一起打包，命名为班级号-学号-姓名-实验5。**