**Java程序设计**

三、问答题

1、简述字符串常量和字符串变量的区别？

答：

简述字符串常量和字符串变量一样，都是一个一维字符数组。两者的区别在于：字符串变量的名字（一维字符数组名）及其所需要的储存空间是显式定义的，并通过名字来引用相应的字符串常量。而字符串变量所需要的储存空间是隐式定义的，并且其根本就没有名字。

2、简述Java中线程的生命周期。

答：

Java中线程的生命周期包括五个状态：

新建状态，就绪状态，运行状态，阻塞状态，销毁状态。

3、简述Java中事件处理模型。

答：

Java中事件处理模型是一种人机交互模型。他有三个基本要素。

1事件源

2事件

3事件监听器

4、简述使用多态的好处。

答：

①可扩充性：多态对代码具有可扩充性。增加新的子类不影响已存在类的多态性，继承性，以及其他特性的运行与操作。

②可替换性：多态对已存在代码具有可替换性。

③灵活性：它在应用中体现了灵活多样的操作，提高了使用效率。

④接口性：多态是超类通过方法签名，向子类提供一个共同的接口，由子类完善或覆盖它而实现。

⑤简化性：多态简化对应用软件的代码编写和修改过程，尤其是在处理大量运算操作时，十分重要。

四、编程题

1、编写一个名为Test1的程序，实现从键盘上输入一个任意字符串，统计这个字符串中的每个字符出现的次数，如：

从键盘上输入：“abcd&+12a12”

则输出：a = 2

b = 1

c = 1

d = 1

& = 1

+ = 1

1 = 2

2 = 2

源程序：

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class Test1 {

static public Map countChar(String s) {

Map<Character, Integer> map = new HashMap<Character, Integer>();

char c = '\0';

for (int i = 0; i < s.length(); i++) {

c = s.charAt(i);

if (map.containsKey(Character.valueOf(c)))

map.put(Character.valueOf(c), Integer.valueOf(map.get(

Character.valueOf(c)).intValue() + 1));

else

map.put(Character.valueOf(c), Integer.valueOf(1));

}

return map;

}

static public void main(String[] string) {

Scanner s = new Scanner(System.in);

String sss = "";

sss = s.nextLine();

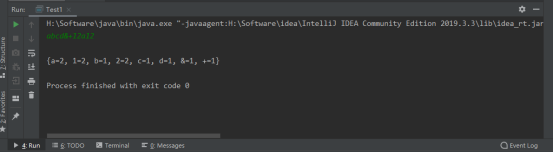
Map map = countChar(sss);

System.out.println(map);

}

}

运行结果截图：



2、编写一个名为Test2的程序，模拟彩票自动选号，运行此程序，可以实现在区间[1,35]中随机产生7个不同的随机整数，并按照从小到大的顺序打印。

源程序：

package test;

public class Test2 {

public static void main(String[] args) {

int ram = 100;

int [] sjs = new int[10];

for(int k =0;k<10;k++){

int m = (int) (Math.random()\*ram+1);

sjs[k] = m;

}

for(int i=0;i<(sjs.length-1); i++ ){

for(int j=i;j<sjs.length;j++ ){

if(sjs[i] > sjs[j]){

int temp = sjs[i];

sjs[i] = sjs[j];

sjs[j] =temp;

}

}

}

for(int l=0;l<10;l++){

System.out.println(sjs[l]);

}

System.out.println("从小到大："+sjs[0]+","

+sjs[1]+","+sjs[2]+","

+sjs[3]+","+sjs[4]+","

+sjs[5]+","+sjs[6]+","

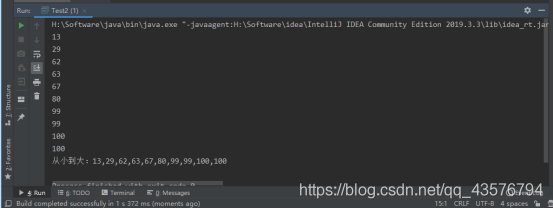
+sjs[7]+","+sjs[8]+","

+sjs[9]);

}

}

运行结果截图：



3、编写一个名为Test3的程序，使用FileInputStream类和FileOutputStream类实现文件的复制。例如：把C盘下的cat.jpg文件复制到D盘下，运行程序时，输入：C:/cat.jpg

D:/cat.jpg

输出结果：拷贝成功！

源程序：

package test;

import java.io.\*;

import java.util.\*;

public class Test3 {

public static void main(String[] args) throws Exception {

String url1 = "H:\\cat.jpg";// 源文件路径

String url2 = "H:\\bat.jpg";// 目标路径（复制到E盘，重命名为b.txt）

copy(url1, url2);

}

private static void copy(String url1, String url2) throws Exception {

FileInputStream in = new FileInputStream(new File(url1));

FileOutputStream out = new FileOutputStream(new File(url2));

byte[] buff = new byte[512];

int n = 0;

System.out.println("=================" + "\n" + "请输入要复制的文件(包含路径)：" + "\n" + url1 + "\n" + "请输入目的地路径和新文件名："

+ "\n" + url2 + "\n" + "=================");

while ((n = in.read(buff)) != -1) {

out.write(buff, 0, n);

}

out.flush();

in.close();

out.close();

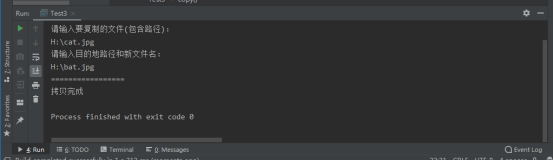
System.out.println("拷贝完成");

}

}

运行结果截图：

c盘无法访问，所有改到H盘



4、编写一个名为Test4的程序，实现把下表中的数据存入一个Map集合中，并分别使用keySet()和entrySet()分别对Map集合进行遍历。

key value

2019001 张三

2019002 李四

2019003 王五

2019004 马六

源程序：

package test;

import java.util.HashMap;

import java.util.Iterator;

import java.util.Map;

import java.util.Set;

public class Text4{

public static void main(String[] args) {

// key为String，value为Object，可放对象等……

Map<String, Object> map = new HashMap<String, Object>();

map.put("2019001","SUM");

map.put("2019002", "徐文静");

map.put("2019003", "王五");

map.put("2019004", "马六");

System.out.println("key value");

for (Map.Entry<String, Object> en : map.entrySet()) {

System.out.println(en.getKey() +","+ en.getValue());

}

}

}

运行结果截图：

