**姓名：丁建文**

**考核题目**

一、选择题 0+2+2+2+2+2+2+0+2+2+2=18

BD

1、下列声明哪个是错误的。（）

A ：int i = 10; B： float f = 1.1; C： double d = 34.4; D： long m = 4990;

B

2、若x是float类变量，x=10/4;则x 的值是（）

A:2 B:2.0 C:2,5 D:编译错误

B

1. float[] f = new float[5];申请了多少字节的内存空间（）

A:25 B:20 C:15 D:10

AC

4、根据下面的代码，int[] s = null;会抛出NullPointerException异常的有（）

A：if( (s!=null) & (s.length()>0) ) B：if( (s!=null) & & (s.length()>0) )  
C：if( (s==null) | (s.length()==0) ) D：if( (s==null) || (s.length()==0) )

AB

1. 下面关于数组说法正确的是（）   
   A）数组实质上是相同类型变量的容器  
   B）创建一个数组必须指定数组元素大小  
   C）char c[]=new char[26];可声明一个含有 26 个元素的 char型数组   
   D）int[] a={1,2,3}; System.out.println(a);可以打印出来1,2,3

C

6、对字符串String str=”abc,ed:,fg,h”;执行这句代码str.split(“:”);会到得到什么结果？

* 1. 得到字符串abc,ed
  2. 得到字符串fg,h
  3. 得到字符串数组[“abc,ed” , ”,fg,h”]
  4. 得到字符“:”

AD

7、关于String 和 StringBuffer 下面说法正确的是 ( )

* 1. String操作字符串不改变原有字符串的内容
  2. StringBuffer连接字符串速度没有String 快 str=”ad”str2=”adf”str+str2
  3. String 可以使用append方法连接字符串 StringBuffer apppend
  4. StringBuffer 在java.util包中 java.lang，不用导包

AB

8、下面说法错误的是（）

* 1. switch case可以使用字符串作为判断条件
  2. String属于基本数据类型
  3. String[]属于引用数据类型
  4. 字符串的初始值为”” String s=null

ABCD

11、下面关于类的描述正确的是（）

A:类中的成员可以相互访问

B:类中定义的属性都有默认值

C:类是具有相同属性和行为的多个对象的统一描述

D:类相当于图纸，对象相当于产品

A

12、下面关于变量的描述正确的是（）

* 1. 成员变量可以不设置初始值
  2. 局部变量可以不设置初始值
  3. 局部变量存在于堆内存中，随着对象的产生而存在，消失而消失
  4. 成员变量存在于栈内存中，随着所属函数或者代码块的运行而存在，结束而释放

BC

13、下面说法正确的是（）

A 异常可以随意的抛出

B NullPointerException和ArrayIndexOutOfBoundsException异常可以不用捕获

C 方法上同时可以声明多个异常类

D throw必须和throws同时使用

二、问答题 3+2+3+2+3+2+2+3+3+2=25

1、请写出基本数据类型

答：Byte short int long float double char boolean

2、char型变量中能不能存贮一个中文汉字?为什么?

答：char的基本数据类型是用来引用单个字符，汉字也是字符的一种，所以能用来存储一个中文汉字，java语言默认的编码格式是unicode，从外向java程序读入文本或者从java程序向外输出文本要注意编码的转换。

3、用最有效率的方法算出2乘以8等於几?

答：2<<3=16;

1. Integer与int的区别

答：Integer是int基本数据类型的封装类，Integer是一个类继承了Object，所以可以使用Object的hashCode（）equals（）等方法，而int只是一个数据，int的默认值为0，而Integer的默认值为null，即Integer可以区分出未赋值和值为0的区别，int则无法表达出未赋值的情况，另外，Integer提供了多个与整数相关的操作方法，例如，将一个字符串转换成整数，Integer中还定义了表示整数的最大值和最小值的常量。

5、抽象类与接口的区别

答：变量：抽象类中的变量可以是普通变量，而接口中的变量只能是静态常量

方法：抽象类中可以有具体的方法，而接口中所有的方法都是抽象方法

继承：一个类只能继承一个抽象类，而可以实现多个接口

关系：抽象类体现的是类与类之间的共性关系，子类和父类存在”is-a”的关系

接口体现的是类的扩展关系，实现类和接口存在”like-a”的关系

1. StringBuffer与String的区别

答：StringBuffer和String都是java用来存储和操作字符串而使用的类；

String对字符串的操作其不可变性很高，任何看起来使字符串发生改变的做法其实都是创建了一个新的字符串，比如String str=”aaa”;str=str+”bbb”；并不是在原有的”aaa”的字符串的基础上连接”bbb”,而是创建了一个新的字符串”aaabbb”,原有的引用变量指向新的字符串，而之前的”aaa”不再有变量引用而被垃圾回收；

StringBuffer就是针对String做改变字符串效率低而提供的类，如果你要求对创建的字符串做更改的操作，最好使用StringBuffer，与String不同的是它对字符串的修改是在原有对象的基础上而没有新建对象，所以StringBuffer修改字符串的效率高得多；

当我们不要求对字符串做修改而只查看时，就使用String，String的不可变性使字符串即使在多线程运行时并发使用而具备很好的安全性。

7、运行时异常与一般异常有何异同？

答：运行时异常是java虚拟机在运行过程中动态发生的异常，与一般异常相比它没有语法错误，所有编译都会都会通过，一般异常不会通过编译；

在throw throws语句中，如果抛出运行时异常，可以不throws声明，调用该方法也可以不捕获，但运行时会在异常出现的地方终端程序；如果抛出一般异常，必须throws，调用该方法也必须捕获try catch。

1. Java中的异常处理机制有哪些（2种）

答：try{}catch{}在try语句块中监视程序是否发生异常，在catch语句块中捕获处理异常

Throw throws通过throw将可能出现的异常向外抛出，throws在方法上声明你要抛出的异常。

1. HashMap是如何保证元素的唯一性的？

答：在HashMap中，键一定是唯一的，存入相同的键的键值对会相互覆盖

如何保证键的唯一性的，我们需要保证键的类覆盖了hashCode()方法和equals()方法，如果两个对象它们hashCode()方法的返回值相等并且equals()方法返回true，那么jvm就会将这两个对象视为相同的对象以保证键的唯一性。

1. 请简述字节流与字符流的区别？

答：字节流和字符流是java语言io流的两个最基本的分支；

字节流的基类是Input/OutputStream,它的实现类有FileInput/OutputStream,BufferedInput/OutputStream等等,它对文件的读写的基本单位是单个字节，可以用来操作图像，音频，视频文件等等；

字符流的基类是Reader/writer,它的实现类有FileReader/Writer,BufferedReader/Writer等等,它对文件的读写的基本单位是单个字符，一般用来操作文本文件；

三、程序问答题 0+3+3+3+3+3+3+3+1+1+3+0+3+1+3+3+3=39

若程序没有错，请写出最终的结果，若有错，请修改

1下面代码打印的结果是？

public static void main(String[] args){

for(int i=0;i<3;i++){

Thread t=new Thread(){

public void run(){

for(int i=0;i<2;i++){ for(int j=0;j<2;j++)

System.out.print(i); j

}

}

};

t.run(); t.start();

}

}

2、 int[] a={1,4,6,7};

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

if(a[i])==7{

continue;

System.out.println(i); 不会被执行的语句，上移一行或者下移一行都可以

}

}

3、final int a = 10;

switch(){ 缺少条件语句的判断表达式

case 0:

int b=10;

System.out.println(a); 加一行break；

case 1:

a=b; final变量不能重新赋值

System.out.println(a);

break;

}

4、

ok:for(int i=0;i<5;i++)

{

for(int j=0;j<5;j++)

{

System.out.println(“i=” + i + “,j=” + j);

if(j == 2){

break ok;

}

}

}

无错误，结果为

i=0,j=0

I=0,j=1

I=0,j=2

5、int a = 10;

float b=a<<1+1\*2;

System.out.println(b);

无错误，结果为

80.0

6、String str=”abcd”;

String c=str.charAt(6); char 0-3之间的数，否则下标越界

System.out.println(c);

7、

class Student{

String name;

int age;

public static void setAge(int age){

age=50;

System.out.println(age);

}

}

class Test{

public static void main(String[] args){

Student.setAge(18);

}

}

程序可以运行，结果是50

但这么写setAge（）方法的形参传入值将没有意义

8

class Fu{

String name;

public Fu(){

name=”yyy”;

}

}

class Zi extends Fu{

String name;

public Zi(String name){

this.name=name;

}

public void introduce(){

System.out.println(name);

}

}

class Test{

public static void main(String[] args){

Zi zi=new Zi(“nnn”);

zi.introduce();

}

}

程序可以运行，结果是nnn

9

interface Ia{

public abstract void println();

void showA();

}

interface Ib extends Ia{

int showB(){}

}

class A implements Ib{ 加上abstract定义为抽象类或者定义具体的方法

int showB(){} 需要被具体实现的方法有3个，println()，showA();showB(){}

}

10

public void calculate(int a,int b){ throws ArithmeticException,MyException

if(b<0){

throw new ArithmeticException(“除数为负数了”);

}

if(a>100){

throw new MyException(“数字太大了”);

}

}

class MyException extends Exception{

public MyException(String msg){

super(msg);

}

}

11

new Thread(new Runnable(){

public void run() throws NullpointerException{ Runnable的run方法不能抛出异常

String[] s=null; try{String[] s=null; s[0]=”a”;}

s[0]=”a”; catch(NullpointerException e){

} e.printStackTrace();}

}).start();

12

ArrayList<Integer> al=new ArrayList<Integer>(20);

for(int i=0;i<30;i++){

al.add(i);

}

Iterator it=al.iterator();

while(it.hasNext()){

Integer i=it.next();

if(i.equals(3)){

al.remove(new Integer(3)); it.remove();

}

}

13

FileReader fr=new FileReader(“D:/abc.txt”);

FileWriter fw=new FileWriter(“D:/a.txt”);

byte[] b=new byte[4]; char [] b=new char[4];

int len=0;

while((len=fr.read())!=-1){ read(b)

fw.write(b,0,len); fw.flush();

} fr.close();fw.close();

14、下面代码的打印结果是：

String str1=”abc”;

String str2=”abc”;

String str3=str1+str2;

String str4=str3;

System.out.println(str1==str2); true

System.out.println(str3==str4); true

System.out.println(str3==(str1+str2)); true

15、两次打印的结果一样吗？

public class Test {

    public static void main(String[] args) {

        char a = '1';

        char b = '2';

        char c = '3';

        System.out.print(a + b + c );

String a1=”1”;

String b1=”2”;

String c1=”3”;

  System.out.print(a1 + b1 + c1 );

    }

}

不一样，第一次会使用ASCII码表字符对应的值相加，得到一个整数，第二次会得到字符串“123”

16、下面的代码打印出什么结果？

public class Test{

public static void main(String[] args) {

Car c = new Car();

}

}

class Car{

static int seat=4;

static{

System.out.println(“一共有”+seat+”个座位”);

}

static int money=100;

}

一共有4个座位

17、下面的代码打印出什么结果？

public class Outer{

内部类是类的成员

private int a=10;

String str=”a”;

class Inner{

int b=20;

String str=”b”;

void print(){

System.out.println(a);//10

System.out.println(str);//b

}

}

}

结果为：

10

b

四、编程题 9+9+9+8+8+4+4+8=59

1. 编写一个Java应用程序，从键盘读取用户输入两个字符串（必须是数字），并重载3个函数分别实现这两个字符串的拼接、整数相加和浮点数相加。对输入的不符合要求的字符串提示给用户，然后重新输入。

import java.util.Scanner;

public class Test{

public static void main(String []args){

while(true){

Scanner input=new Scanner(System.in);

System.out.println(“请输入两个数字”);

try{

float a=input.nextFloat();

float b=input.nextFloat();

System.out.println(“1->拼接2->整数相加3->浮点数相加”);

switch(input.nextInt()){

case 1:

append(a,b);

break;

case 2:

plusInt(a,b);

break;

case 3:

plusFloat(a,b);

break;

default:

break;

}

}catch(Exception e){

e.printStackTrace();

}

}

}

private static void append(float a,float b){

System.out.println(a+””+b);

}

private static void plusInt(float a,float b){

System.out.println((int)a+(int)b);

}

private static void plusFloat(float a,float b){

System.out.println(a+b);

}

}

1. 有一个学生数组stu[] s={学生1，学生2，...};按照学生的年龄和姓名进行自然排序。（提示：使用Arrays.sort()，具体看jdk文档）

import java.util.Arrays;

import java.util.Comparator;

public class Test{

public static void main(String []args){

Student [] students=new Student []{new Student(“张三”,18),new Student(“李四”,17),new Student(“王五”,19)};

Arrays.sort(students,new Comparator<Student>(){

public int compare(Student s1,Student s2){

if(s1.age==s2.age){return s1.name.compareTo(s2.name);}

return s1.age-s2.age;

}});

}

}

class Student{

String name;

int age;

public Student(String name,int age){

this.name=name;

this.age=age;

}

}

3、

String url="http://120.32.4.343:8080/login.jsp?name=yuann&&password=123456&&sex=men&&login=1470277690256";（login后面是毫秒值）

对请求地址进行截取，输出登录用户的信息（用户名，当前登录时间（需要格式化时间））

打印结果 姓名：yuann 登录时间：2016/08/29

import java.text.SimpleDateFormat;

public class Test{

public static void main(String [] args){

String url="http://120.32.4.343:8080/login.jsp?name=yuann&&password=123456&&sex=men&&login=1470277690256";

String [] cut1=url.split(“jsp?”);

String [] cut2=cut1[1].split(“&&”);

String [] nameMap=cut2[0].split(“=”);

String name=nameMap[1];

String [] loginMap=cut2[3].split(“=”);

String login=loginMap[1];

Long time=Long.parseLong(login);

SimpleDateFormat sdf=new SimpleDateFormat(“yyyy/MM/dd”);

String timeStr=sdf.format(time);

System.out.println(“姓名：”+name+”登录时间：”+timeStr);

}

}

1. 数据库中保存着飞机票（班次、票价、时间）
2. 用户输入班次，则返回指定的飞机票
3. 用户输入票价，则按票价低到高的顺序返回飞机票，对选出的飞机票迭代输出
4. 用户输入时间，则按时间先后顺序返回飞机票，对选出的飞机票迭代输出

提示：使用TreeSet集合保存飞机票，第2、3小问可以使用我们学过的迭代来做，也可以使用tailSet()方法，具体查看jdk文档

import java.util.\*;

public class Test{

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

while(true){

System.out.println("1->查询班次2->查询票价3->查询时间");

switch(input.nextInt()){

case 1:

searchOrder(input);

break;

case 2:

searchPrice(input);

break;

case 3:

searchTime(input);

break;

default:

break;

}

}

}

private static void addIntoSet(TreeSet<AirTicket> ts){

ts.add(new AirTicket("MU3358",550,"07:35"));

ts.add(new AirTicket("MU3729",850,"04:49"));

ts.add(new AirTicket("MU6668",350,"06:35"));

ts.add(new AirTicket("MF2558",490,"17:25"));

ts.add(new AirTicket("MF3394",1050,"14:55"));

ts.add(new AirTicket("ME9758",1550,"19:35"));

ts.add(new AirTicket("ME4258",3000,"22:06"));

}

private static void searchOrder(Scanner input){

System.out.println("输入班次:");

String airOrder=input.next();

TreeSet<AirTicket> ts=new TreeSet<AirTicket>();

addIntoSet(ts);

for(AirTicket airTicket:ts){

if(airOrder.equals(airTicket.airOrder)){

System.out.println(airTicket);

break;

}

}

}

private static void searchPrice(Scanner input){

TreeSet<AirTicket> ts=new TreeSet<AirTicket>(new Comparator<AirTicket>(){

public int compare(AirTicket a1,AirTicket a2){

if(a1.price==a2.price){

return a1.airOrder.compareTo(a2.airOrder);

}

return a1.price-a2.price;

}

});

addIntoSet(ts);

System.out.println("输入票价");

int price=input.nextInt();

Set<AirTicket> tailts=ts.tailSet(new AirTicket("",price,""));

for(AirTicket airTicket:tailts){

System.out.println(airTicket);

}

}

private static void searchTime(Scanner input){

TreeSet<AirTicket> ts=new TreeSet<AirTicket>(new Comparator<AirTicket>(){

public int compare(AirTicket a1,AirTicket a2){

if(a1.time.equals(a2.time)){

return a1.airOrder.compareTo(a2.airOrder);

}

return a1.time.compareTo(a2.time);

}

});

addIntoSet(ts);

System.out.println("输入时间");

String time=input.next();

Set<AirTicket> tailts=ts.tailSet(new AirTicket("",0,time));

for(AirTicket airTicket:tailts){

System.out.println(airTicket);

}

}

}

class AirTicket implements Comparable<AirTicket>{

String airOrder;

int price;

String time;

public AirTicket(String airOrder,int price,String time){

this.airOrder=airOrder;

this.price=price;

this.time=time;

}

public int compareTo(AirTicket airTicket){

return airOrder.compareTo(airTicket.airOrder);

}

public String toString(){

return "班次："+airOrder+"--票价："+price+"--时间："+time;

}

}

1. 使用HashMap和HashSet保存全国的省份和城市，例如：[浙江省, {杭州、温州、宁波} ] 、[广东省, {广州、深圳、珠海} ]，城市需要封装成一个类，属性包括名称和是否是省会城市。需求：

1、用户输入浙江省，返回省内的城市，并标注出省会城市

2、用户输入一个城市，可以显示在哪个省份。

import java.util.\*;

class City{

private String name;

private boolean isCapital;

public City(String name,boolean isCapital){

this.name=name;

this.isCapital=isCapital;

}

public String getName(){

return name;

}

public int hashCode(){

return name.hashCode();

}

public boolean equals(Object o){

if(o instanceof City){

City c=(City)o;

if(name.equals(c.name)){

return true;

}

}

return false;

}

public String toString(){

if(isCapital){

return name+"\*";

}

return name;

}

}

public class Test{

public static void main(String[] args) {

HashMap<String,HashSet<City>> hm=new HashMap<String,HashSet<City>>();

HashSet<City> zhejiangProvince=new HashSet<City>();

zhejiangProvince.add(new City("杭州",true));

zhejiangProvince.add(new City("温州",false));

zhejiangProvince.add(new City("宁波",false));

HashSet<City> guangdongProvince=new HashSet<City>();

guangdongProvince.add(new City("广州",true));

guangdongProvince.add(new City("深圳",false));

guangdongProvince.add(new City("珠海",false));

hm.put("浙江省",zhejiangProvince);

hm.put("广东省",guangdongProvince);

Scanner input=new Scanner(System.in);

while(true){

System.out.println("1->查询省份2->查询城市");

switch(input.nextInt()){

case 1:

searchProvince(hm,input);

break;

case 2:

searchCity(hm,input);

break;

default:

break;

}

}

}

private static void searchProvince(HashMap<String,HashSet<City>> hm,Scanner input){

System.out.println("输入省份");

String key=input.next();

HashSet<City> value=hm.get(key);

for(City city:value){

System.out.println(city);

}

}

private static void searchCity(HashMap<String,HashSet<City>> hm,Scanner input){

System.out.println("输入城市");

String cityName=input.next();

Set<Map.Entry<String,HashSet<City>>> set=hm.entrySet();

ok:for(Map.Entry<String,HashSet<City>> entry:set){

String province=entry.getKey();

HashSet<City> hs=entry.getValue();

for(City city:hs){

if(city.getName().equals(cityName)){

System.out.println(province);

break ok;

}

}

}

}

}

1. 将指定文件夹下面的所有txt文件筛选出来，并且将内容合并到同一个文件中，注意编码转换的问题。

import java.io.\*;

public class Test{

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

File file=new File("e:/abc");

File [] files=file.listFiles(new FilenameFilter(){

public boolean accept(File dir,String name){

return name.endsWith(".txt");

}

});

InputStreamReader isr=null;

OutputStreamWriter osw=new OutputStreamWriter(new FileOutputStream("e:/combine.txt"),"utf-8");

for(File f:files){

isr=new InputStreamReader(new FileInputStream(f),"utf-8");

char [] c=new char[10];

int len=0;

while((len=isr.read(c))!=-1){

osw.write(c,0,len);

}

}

isr.close();

osw.close();

}

}

1. 对比两个文本文件的内容，将每行文字内容不相同的位置找出来，并保存的另一个文本中（行号、不相同的文字是哪两个）例如：

行号 原 新

3 我 你

import java.io.\*;

public class Test{

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

LineNumberReader lmr1=new LineNumberReader(new FileReader("e:/a.txt"));

LineNumberReader lmr2=new LineNumberReader(new FileReader("e:/b.txt"));

BufferedWriter bw=new BufferedWriter(new FileWriter("e:/ab.txt"));

String line1;

String line2;

int lineNum1;

int lineNum2;

bw.write("行号\t原\t新\t");

bw.newLine();

while((line1=lmr1.readLine())!=null|(line2=lmr2.readLine())!=null){

lineNum1=lmr1.getLineNumber();

lineNum2=lmr2.getLineNumber();

if(line1!=null&&line2!=null){

if(!line1.equals(line2)){

int lineNum=lineNum1>=lineNum2?lineNum1:lineNum2;

System.out.println(lineNum+"\t"+line1+"\t"+line2);

bw.write(lineNum+"\t"+line1+"\t"+line2);

bw.newLine();

}

}else{

int lineNum=lineNum1>=lineNum2?lineNum1:lineNum2;

System.out.println(lineNum+"\t"+line1+"\t"+line2);

bw.write(lineNum+"\t"+line1+"\t"+line2);

bw.newLine();

}

}

lmr1.close();

lmr2.close();

bw.close();

}

}

注:对于行号，可以使用LineNumberReader，请查看jdk文档，也可以使用已学过的计数器来实现。

8、思考题：写一个加密算法，对一个字符串进行加密，并将密文保存到文件中，然后再加密文解密，输出到控制台。（加密的算法自己定，例如使用异或就可以实现一个简单的加密）

import java.io.\*;

public class Test{

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

String src="飞天小女警";

byte [] b=src.getBytes();

FileOutputStream fos=new FileOutputStream("e:/8t.txt");

for(int i:b){

fos.write(i^0xff);

}

fos.close();

FileInputStream fis=new FileInputStream("e:/8t.txt");

byte [] buf=new byte[fis.available()];

int index=0;

byte a=0;

while((a=(byte)fis.read())!=-1){

buf[index]=(byte)(a^0xff);

index++;

}

System.out.println(new String(buf));

}

}