**2020年上海市高等学校信息技术水平考试试卷**

**二级 Java程序设计（A 场）**

（本试卷考试时间 120 分钟）

**一、单选题 ( 本大题 10 道小题 ，每小题 1.5 分，共 15 分），从下面题目给出的A、B、C、D四个可供选择的答案中选择一个正确答案。**

1.下列不合法的赋值语句是\_\_\_\_\_\_\_\_\_\_。

A.int x=12345;

B.int x=12345L;

C.double x=3.14;

D.float x=(float)3.1415;

2.以下语句序列执行后的输出结果是\_\_\_\_\_\_\_\_\_\_\_\_。

String s1 = new String(”Hello”);

String s2 = new String(”Hello”);

System.out.println(s1 == s2);

System.out.println(s1.equals(s2));

A.true

false

B.Hello

true

C.Hello

Hello

D.false

true

3.执行下面程序段后，i的值为\_\_\_\_\_\_\_\_\_。

int i;

for (i=0;i<10;i++)

if(i==3 ) break;

A.0

B.3

C.9

D.10

4.用static修饰的方法被称为\_\_\_\_\_\_\_\_。

A.类方法

B.构造方法

C.实例方法

D.抽象方法

5.若对成员变量i进行如下的定义：

int i=10;

则关于i的正确说法是\_\_\_\_\_\_\_\_。

A.i能被同一个包中的其他类或对象访问

B.i不能被同一个类中所有对象访问

C.i只能被本类中的对象访问

D.i能被其他包中同父类的子类对象访问

6.下面会导致死循环的语句是\_\_\_\_\_\_\_\_\_\_。

A.for(int k=0; k<0; k++){ System.out.println("a");}

B.for(int k=10; k>0; k--){ System.out.println("a");}

C.for(int k=0; k<10; k--){ System.out.println("a");}

D.for(int k=0; k>0; k++){ System.out.println("a");}

7.在下列有关Java语言特点的叙述中，错误的选项是\_\_\_\_\_\_\_\_\_\_。

A.Java是面向过程的编程语言

B.Java支持分布式计算

C.Java是跨平台的编程语言

D.Java支持多线程

8.\_\_\_\_\_\_\_\_\_\_\_\_\_\_不是java.awt组件。

A.Label

B.JButton

C.Button

D.TextArea

9.关于异常处理，如果程序中要抛出异常，应使用\_\_\_\_\_\_\_\_\_\_\_语句。

A.catch

B.throw

C.finally

D.throws

10.下列程序中变量c的结果为\_\_\_\_\_\_\_\_\_\_\_。

public class test{

public static void main(String args[]) {

int a=10;

int b;

int c;

if(a>50) {

b=9;

}

c=b+a;

}

}

A.10

B.0

C.19

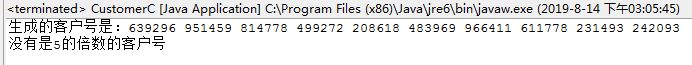
D.编译出错

**二、程序填空题 ( 本大题 2 道小题 ，每空 3 分，共 24 分）。**

1. 本程序的功能如下：为10位客户生成长度为6位的客户号,然后统计客户号是5的倍数的个数并输出这些客户号；若没有产生是5的倍数的客户号，则输出其相应信息。下面两图所示分别是程序某两次运行结果。请将程序补充完整。



或



**public** **class** CustomerC {

**public** **static** **void** main(String[] args) {

**int** count = 0; // 存放满足要求的客户数

String s = "";

**int** CustN[] = \_\_\_\_\_(1)\_\_\_\_\_\_; // 定义存放客户号的数组

System.***out***.print("生成的客户号是：");

**for** (**int** i = 0; i < 10; i++) {

CustN[i] = \_\_\_\_\_(2)\_\_\_\_\_ (Math.*random*() \* 900000 + 100000);

System.***out***.print(CustN[i] + " ");

**if** (CustN[i] % 5 == \_\_\_\_(3)\_\_\_\_) {

\_\_\_\_(4)\_\_\_\_;

s = s + CustN[i] + " ";

}

}

System.***out***.println();

**if** (count > 0)

System.***out***.println("是5的倍数的客户号有" + count + "个：" + s);

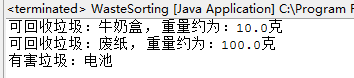
**else**

System.***out***.println("没有是5的倍数的客户号");

}

}

2. 以下是为垃圾回收利用而设计的一段程序，其中定义了一个抽象类Waste，包含成员变量和抽象成员方法droplitter()（见程序的注解）；另外分别定义Waste的子类RecyclableWaste和HazardousWaste，在主程序中生成对象并使用成员方法，程序正常运行时的输出界面如下图所示，请将程序补充完整。



**abstract** **class** Waste {

String name; // 垃圾名

**double** weight = 10.0; // 垃圾重量

\_\_\_\_\_(1)\_\_\_\_\_ // 定义droplitter()成员方法

}

**class** RecyclableWaste **extends** Waste { // 可回收垃圾

**public** RecyclableWaste(String n, **double** w) { // 可回收垃圾

name = n;

\_\_\_\_\_(2)\_\_\_\_\_

}

**public** RecyclableWaste(String name) { // 构造方法

**this**(name, 10.0);

}

**void** droplitter() {

System.***out***.println("可回收垃圾：" + name + "， 重量约为：" + weight + "克");

}

}

**class** HazardousWaste \_\_\_\_\_(3)\_\_\_\_\_ Waste { // 有害垃圾

**public** HazardousWaste(String n) {

name = n;

}

**void** droplitter() {

System.***out***.println("有害垃圾：" + name);

}

}

**public** **class** WasteSorting {

**public** **static** **void** main(String args[]) {

RecyclableWaste reW1 = **new** RecyclableWaste("牛奶盒");

reW1.droplitter();

RecyclableWaste reW2 = **new** RecyclableWaste(\_\_\_\_\_(4)\_\_\_\_\_);

reW2.droplitter();

HazardousWaste haW = **new** HazardousWaste("电池");

haW.droplitter();

}

}

**三、操作题**

****

**注意：**

**1.所有例子的图示都以Eclipse运行环境为例，在此仅作参考。**

**2.编程所需的"常用方法参考"见操作题最后。**

（一）程序调试题(本大题 3 小题，共10个错误，每改正一个3分，共 30 分）

**调试题基本要求：不增加或删除行，按题中的要求调试修改程序，并将结果（包括源文件、字节码文件）保存到C:\KS目录中。请务必在所修改的语句后加上注释语句："//\*\*\*\*\*"。**

1．从"C:\素材"中取出GC1.java文件，文件存在3个错误。此程序模拟某种棋类训练方式:当一选手连赢三次即可获得升级；程序中用产生随机数的方法来表示每一次的博弈结果当随机数大于0.5显示为"T"，小于0.5则为"F"，一旦连续出现三次"T"，即表示该选手可获得升级。下图为程序的某一次正确运行结果。



**public** **class** GC1 {

**public** **static** **void** main(String[] args) {

**int** c;

**double** d;

**int** i = 0;

**while** (i < 3) {

d = *random*();

**if** (d > 0.5) {

i++;

c = 'T';

} **else** {

i = 0;

c = 'F';

}

System.***out***.print(c + " ")

}

System.***out***.println();

System.***out***.print("恭喜 ！连赢三次，获得升级。 ");

}

}

2．从"C:\素材"中取出GC2.java文件，该程序有4个错误。此文件中定义了一个接口Instrument(乐器)，其中包含一个方法play（演奏），NationalInstrument类实现了这个接口，在GC2的main方法中创建了相应的对象并完成相应的功能。程序正确运行结果如下图所示：



**interface** Instrument {

**void** play(String musicName);

}

**class** NationalInstrument **extends** Instrument {

**private** String instrumentName;

NationalInstrument(String instrumentName) {

**this**.instrumentName = instrumentName;

}

NationalInstrument() {

}

**public** String getInstrumentName() {

**return** instrumentName;

}

**public** **void** setInstrumentName(String instrumentName) {

instrumentName = instrumentName;

}

**public** **void** play(**int** musicName) {

System.out.println***(in***strumentName + ":" + musicName);

}

}

**public** **class** GC2 {

**public** **static** **void** main(String[] args) {

NationalInstrument instrument1 = new NationalInstrument();

instrument1.setInstrumentName("琵琶");

instrument1.play(十面埋伏);

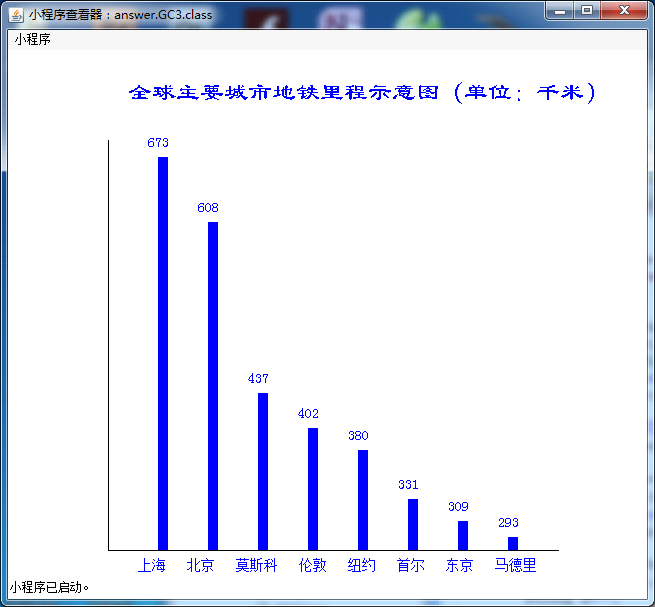
NationalInstrument instrument2 = new NationalInstrument("二胡");

instrument2.play("二泉映月");

}

}

3．截至2018年，全球主要城市地铁里程分别为（单位：千米）：上海：673，北京：608，莫斯科：437，伦敦：402，纽约：380 ，首尔：331，东京：309，马德里：293；在"C:\素材"中的GC3.java程序可绘制如下图所示的全球主要城市地铁里程示意图，但程序中有3处错误，请予以改正。



**import** java.applet.\*;

**import** java.awt.\*;

**public** **class** GC3 **extends** Applet{

String[] xPoints = {"上海","北京","莫斯科","伦敦","纽约","首尔","东京","马德里"};

**int** yuanPoints = {673,608,437,402,380,331,309,293};

String yPoints ;

**public** **void** paint( Graphics) {

g.drawLine(100,500,550,500);//绘制横轴

g.drawLine(100,500,100,90);//绘制纵轴

g.setColor(Color.***blue***);

Font font=**new** Font("隶书",Font.***PLAIN***,24);

g.setFont(font);

g.drawString("全球主要城市地铁里程示意图（单位：千米）", 120, 50);

Font font1=**new** Font("宋体",Font.***PLAIN***,14);

g.setFont(font1);

**for** (**int** i=0;i<=7;i++){

g.drawRect(50\*i+150,780-yuanPoints[i],10,yuanPoints[i]-280);

yPoints=""+yuanPoints[i];

g.drawString(yPoints,50\*i+140, 770-yuanPoints[i]);

}

Font font2=**new** Font("宋体",Font.***PLAIN***,14);

g.setFont(font2);

String s="";

**for** (**int** i=0;i<=7;i++)

s+=xPoints[i]+" ";

g.drawString(s,130, 520);

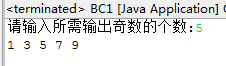
}

}

（二）编程题(本大题共3题任选2题完成，多做不加分，每题17.5分，共 35 分）

**编程题基本要求：程序完成后将程序编辑及运行所需要的所有文件（包括源程序文件、字节码文件）存放到C:\KS目录下。**

1．从"C:\素材"中取出BC1.java文件，在该文件基础上进行补充，完成程序中printodds方法的编写。该程序实现由键盘输入一自然数，程序输出由1开始的相应个数的奇数序列，程序运行结果如下图所示。



**import** java.io.\*;

**public** **class** BC1 {

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

BufferedReader buf;

buf = **new** BufferedReader(**new** InputStreamReader(System.***in***));

System.***out***.print("请输入所需输出奇数的个数:");

**int** n = Integer.*parseInt*(buf.readLine());

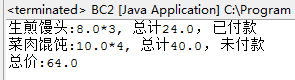
*printodds*(n);

}

//在此完成printodds方法的编写

}

2．从"C:\素材"中取出应用程序BC2.java文件，在该文件基础上完成程序的编写。该程序的功能是创建两个WebOrder对象并调用相应的方法，实现网络订餐的信息设置和显示。WebOrder类定义了四个成员变量（见程序中的注解），下图是程序正确运行后显示的执行结果。



**class** WebOrder {

**private** String dishName; // 货名

**private** **int** quantity; // 数量

**private** **double** price; // 单价

**private** **boolean** isPaid; // 付账状态："true"为已付账

**public** WebOrder(String dishName, **int** quantity, **double** price) {

**this**.dishName = dishName;

**this**.quantity = quantity;

**this**.price = price;

}

**public** **void** addDish(**int** numToAdd) { //为对象添加份数

//在此处定义方法体

}

**public** **void** setIsPaid(**boolean** paid) { //为对象设置付账信息

//在此处定义方法体

}

**public** **double** getPrice() { //获取对象的总价

//在此处定义方法体

}

**public** String getDishName() {//获取对象的货名

//在此处定义方法体

}

**public** String toString() { //获取对象的相关信息

//在此处定义方法体

}

}

**public** **class** BC2 {

**public** **static** **void** main(String[] args) {

WebOrder o1, o2;

o1 = **new** WebOrder("生煎馒头", 1, 8);

o2 = **new** WebOrder("菜肉馄饨", 1, 10);

o1.addDish(1);

o2.addDish(3);

o1.addDish(1);

o1.setIsPaid(**true**);

o2.setIsPaid(**false**);

System.***out***.println(o1.toString());

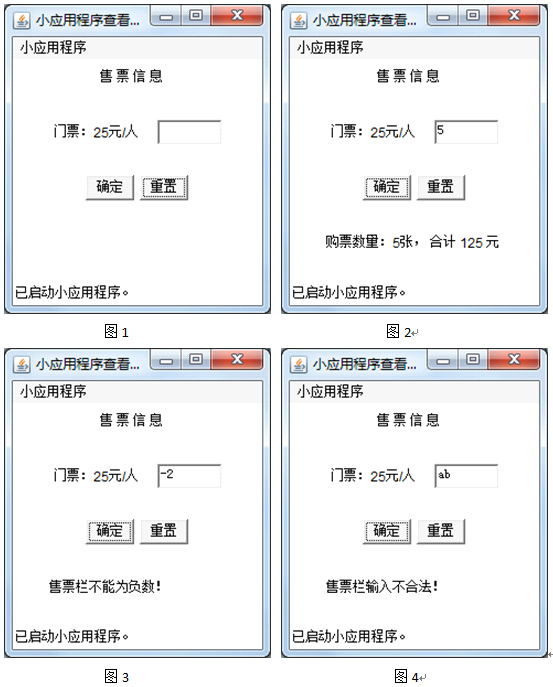
System.***out***.println(o2.toString());

System.***out***.println("总价:"+(o1.getPrice()+o2.getPrice()));

}

}

3．从"C:\素材"中取出BC3.java文件，该程序实现基本的售票功能。请在原有文件基础上完成程序的编写。下图1为程序的初始界面；当在文本域中输入数据并单击"确定"按钮后，程序要对输入数据进行判断，当输入合法时，在界面最下方的标签信息中显示购票数量及合计金额，如图2所示；当输入为小于0的整数时，标签信息中显示"售票栏不能为负数！"，如图3所示；当输入为非法数字时，在标签中显示"售票栏输入不合法！"，如图4所示；单击"重置"按钮后，程序则将文本域和最下方标签中的内容清空。



**import** java.awt.\*;

**import** java.awt.event.\*;

**import** java.applet.\*;

**public** **class** BC3 **extends** Applet **implements** ActionListener {

Label ja = **new** Label(" 售 票 信 息 ");

Label jb = **new** Label("门票：25元/人");

TextField t1 = **new** TextField(5);

Button b1 = **new** Button(" 确定 ");

Button b2 = **new** Button(" 重置 ");

Label jc = **new** Label(" ");// 用于显示结果

Panel p1 = **new** Panel();

Panel p2 = **new** Panel();

Panel p3 = **new** Panel();

Panel p4 = **new** Panel();

**public** **void** init() {

setLayout(**new** GridLayout(4, 1));// 设置界面布局

p1.add(ja);

p2.add(jb);

p2.add(t1);

p3.add(b1);

p3.add(b2);

p4.add(jc);

add(p1);

add(p2);

add(p3);

add(p4);

b1.addActionListener(**this**);

b2.addActionListener(**this**);

}

// 实现ActionListener接口中的actionPerformed()方法

**public** **void** actionPerformed(ActionEvent e) {

//在此完成对应程序代码的编写

}

}

**常用方法参考：**

**String**

public int length()

public char charAt(int number)

public int compareTo(String anotherString)

public boolean equals(String comparison)

public String valueOf(double number)

public String valueOf(int number)

**Math**

public static double sqrt(double number)

public static double random()

**Integer**

public static int parseInt (String s)

**Double**

public static double parseDouble(String s)

**Graphics class**

void setColor(Color color)

void setFont(Font font)

void drawRect(int x, int y, int width, int height)

void fillRect(int x, int y, int width, int height)

void drawOval(int x, int y, int width, int height)

void fillOval(int x, int y, int width, int height)

void setBounds(int a, int b, int width, int height)

void drawLine(int x1, int y1, int x2, int y2)

void drawString(String, int x, int y)

**Label and Button and TextField classes**

void addActionListener(ActionListener listener)

String getText()

void setText(String str)

**Choice class**

void addItemListener(ItemListener listener)

String getSelectedItem( )

**Applet class**

void destroy()

void paint(Graphics g)

void init()

void repaint()

void start()

void stop()

**ActionListener interface**

void actionPerformed(ActionEvent e)

**ItemListener interface**

void itemStateChanged(ItemEvent e)

**MouseListener interface**

void mousePressed(MouseEvent e)

void mouseReleased(MouseEvent e)

void mouseClicked(MouseEvent e)

void mouseEntered(MouseEvent e)

void mouseExited(MouseEvent e)

**Mouse class**

int getX()

int getY()

void addMouseListener(MouseListener listener)

**Object class**

Object getSource()