

2020 年 10 月高等教育自学考试全国统一命题考试

Java 语言程序设计(一) 试卷

(课程代码 04747)

本试卷共 9 页,满分 100 分,考试时间 150 分钟。

考生答题注意事项:

1. 本卷所有试题必须在答题卡上作答。答在试卷上无效,试卷空白处和背面均可作草稿纸。
2. 第一部分为选择题。必须对应试卷上的题号使用 2B 铅笔将“答题卡”的相应代码涂黑。
3. 第二部分为非选择题。必须注明大、小题号,使用 0.5 毫米黑色字迹签字笔作答。
4. 合理安排答题空间,超出答题区域无效。

第一部分 选择题

一、单项选择题:本大题共 10 小题,每小题 1 分,共 10 分。在每小题列出的备选项中只有一项是最符合题目要求的,请将其选出。

1. Java 语言中不支持的语法成分或功能是

- | | |
|-------|------------|
| A. 引用 | B. #define |
| C. 继承 | D. new |

2. 某函数中定义了 4 个变量: int iVal、long lVal、float fVal 及 double dVal,且已经赋了初值。则下列选项中,编译出现错误的是

- | | |
|-----------------|-----------------|
| A. lVal = fVal; | B. fVal = iVal; |
| C. fVal = lVal; | D. dVal = fVal; |

3. 以下程序代码段的输出结果是

```
int sumn = 0, sumk = 0, n, k;  
for (n = 0, k = 0; k < 8; n++, k++) { sumn += n; sumk += k; }  
System.out.print(sumn+" "+sumk);
```

- | | |
|---------|----------|
| A. 28 8 | B. 28 28 |
| C. 8 28 | D. 8 8 |

4. 下列对类的静态成员变量的描述中,正确的是

- A. 不能通过类的对象访问类的静态成员变量
- B. 类的每个对象都有自己的静态成员变量
- C. 类的不同对象有不同的静态成员变量值
- D. 静态成员变量是类的所有对象共享的数据

5. 调用方法时,使用值传递方式,形参是引用。下列叙述中正确的是

- | | |
|-----------------|-----------------|
| A. 形参得到实参所指对象的值 | B. 实参得到形参所指对象的值 |
| C. 形参和实参指向同一地址 | D. 形参和实参无联系 |

6. 以下程序代码段的输出结果是

```
int intArray[] = {2, 4, 6, 8};  
int result = 1, i = 1;  
while(i < 3) result = result * intArray[i++];
```

System.out.print(result);

- | | |
|-------|--------|
| A. 8 | B. 24 |
| C. 48 | D. 192 |

7. 设引用类型的变量 p 标记为 final,则下列叙述中正确的是

- A. p 不能再指向其他对象,但它所指对象中的属性值可以改变
- B. p 不能再指向其他对象,它所指对象中的属性值也不可以改变
- C. p 能指向其他对象,它所指对象中的属性值也可以改变
- D. p 能指向其他对象,但它所指对象中的属性值不可以改变

8. 以下程序代码段的输出结果是

```
File f = new File("/home/mydic/d07.Java");  
System.out.println(f.getName());
```

- | | |
|-------------------------|-------------------------|
| A. /home/mydic/d07.Java | B. \home\mydic\d07.Java |
| C. d07 | D. d07.Java |

9. 下列关于 Swing 组件与 AWT 组件的叙述中,正确的是

- A. Swing 组件在实现时不包含任何本地代码
- B. AWT 组件在实现时不包含任何本地代码
- C. AWT 组件比 Swing 组件的功能更强
- D. Swing 组件在不同的平台下运行会有差异

10. API 中线程操作方法 yield()的作用是

- | | |
|--------------|-----------------|
| A. 启动新建线程对象 | B. 定义线程对象要执行的操作 |
| C. 强制终止线程的执行 | D. 测试当前线程是否在活动 |

第二部分 非选择题

二、填空题：本大题共 10 小题，每空 2 分，共 20 分。

- 若执行程序 HelloWorldApp.java 时键入的命令是 java HelloWorldApp arg1 arg2，则程序中语句 System.out.print(args[1]);的作用是_____。
- 导致程序运行时出现 ArrayIndexOutOfBoundsException 异常的错误原因是_____。
- 设有表示人名的类 Name，其中含有两个 String 类型的数据成员变量 fn 和 ln，分别表示名字中的名和姓。为类 Name 添加一个构造方法，形参分别是 na 和 fna，请写出此构造方法：_____。
- 同一个类中多个方法有相同的名字及不同的参数列表，这种情况称为_____。
- 已知有字符串 s，将 s 中出现的所有“tian”改为“yue”并赋给字符串 t 的语句是_____。
- 用户定义了一个不能被继承的类，在这个类的声明中使用的关键字必须含有_____。
- 在关闭一个缓冲区输出流之前，为了强制输出剩余数据，一般应先使用_____方法。
- 顶层容器获得其默认的内容窗格的方法是_____。
- 创建一个 3 行、20 列、显示初始字符串“Input Lines”的文本区 textA 的语句是_____。
- Java 中，为阻止多个线程同时访问同一个条件变量，可以为每一个对象实例配有一个_____。

三、简答题：本大题共 5 小题，每小题 4 分，共 20 分。

- 写出基本数据类型 byte、boolean、char 和 short 分别占用的字节数。
- 设有整数 x 和 y，两者关系如下。使用一个 if 语句，根据 x 的值计算 y 的值。

$$y = \begin{cases} x - 1 & \text{如果 } x \text{ 是正奇数} \\ x / 2 & \text{如果 } x \text{ 是正偶数} \\ x + 1 & \text{如果 } x \text{ 不是正数} \end{cases}$$

- 列举静态方法的特点。
- 简要叙述接口与多重继承的关系。
- Java 的抢占式线程调度中，当前运行的线程在何种情况下不再占有 CPU？

四、程序填空题：本大题共 3 小题，每空 2 分，共 18 分。

- 设有二维数组 A 和 B，以下程序将矩阵 X 保存在数组 A 中，数组 B 中保存如矩阵 Y 所示的内容。

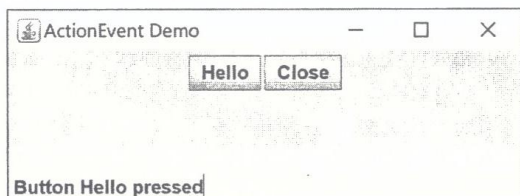
$$X = \begin{bmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \\ 51 & 52 & 53 & 54 \end{bmatrix} \quad Y = \begin{bmatrix} 54 & 44 & 34 & 24 & 14 \\ 53 & 43 & 33 & 23 & 13 \\ 52 & 42 & 32 & 22 & 12 \\ 51 & 41 & 31 & 21 & 11 \end{bmatrix}$$

```
public class Test26{
    public static void main(String[] args) {
        int A[ ][ ] = new int[5][4], B[ ][ ] = new int[4][5]; int i, j;
        for(i = 0; i < 5; i++)
            for(j = 0; ①; j++) A[i][j] = ②;
        for(i = 0; i < 4; i++)
            for(j = 0; j < 5; j++) ③;
    }
}
```

- 以下程序将输出：Jim 2019100 [Internet project]。

```
import java.util.Vector;
class Person {
    private String name = "John";
    public String getName(){ return name; }
    public void setName(String n){ name = n; }
}
class Employee extends Person {
    private int employeeNumber;
    public int getEmployeeNumber(){ return employeeNumber; }
    public void setEmployeeNumber(int number){ employeeNumber = number; }
}
class Manager extends Employee {
    private String department;
    public Vector<String> responsibilities;
    public Vector<String> getResponsibilities(){ return responsibilities; }
}
public class Test27{
    public static void main(String [] args) {
        Manager jim = new Manager();
        jim.setName("Jim"); jim.setEmployeeNumber (2019100);
        jim.responsibilities = ①;
        jim.responsibilities.add("Internet project");
        System.out.print(jim.②+" ");
        System.out.print(jim.getEmployeeNumber()+" ");
        System.out.println( ③ );
    }
}
```

28. 以下程序显示两个按钮和一个文本区，并响应按钮事件。当按下按钮 Hello 时，在文本区显示“Button Hello pressed”，如题 28 图所示；当按下按钮 Close 时，关闭窗口。



题 28 图

```
import java.awt.*; import javax.swing.*; import java.awt.event.*;
public class Test28{
    public static void main(String[] args){
       (ActionEventDemo aed = new(ActionEventDemo());
        aed.go();
    }
}
class(ActionEventDemo implements ① {
    JFrame frame; JButton b1,b2; JTextArea ta; JPanel panel;
    public void go(){
        frame = new JFrame("ActionEvent Demo");
        b1 = new JButton("Hello"); b2 = new JButton("Close");
        ta = new JTextArea(0, 30);
        panel = new JPanel(); panel.add(b1); panel.add(b2);
        b1.addActionListener(this); b2.addActionListener(this);
        frame.getContentPane().add(panel, BorderLayout.CENTER);
        frame.getContentPane().add(ta, BorderLayout.SOUTH);
        frame.pack(); frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    public void actionPerformed(ActionEvent e) {
        JButton mybut = (JButton)e.②;
        if ( mybut ③ ){
            ta.append("\n Button Hello pressed");
        }
        else System.exit(0);
    }
}
```

五、程序分析题：本大题共 3 小题，每小题 6 分，共 18 分。

29. 阅读以下程序，写出该程序的输出结果。

```
abstract class ObjectStorage{
    int objectnum=0;
    int storage[] = new int[100];
    abstract void put(int o);
    abstract int get();
}
class Stack extends ObjectStorage{
    private int point=0;
    public void put(int o){
        storage[point++]=o;
        objectnum++;
    }
    public int get(){
        objectnum--;
        System.out.println(storage[--point]);
        return storage[point];
    }
}
public class Test29{
    public static void main(String args[]){
        Stack s = new Stack();
        int initV = 5;
        s.put(initV * 2); s.put(initV * 3);
        s.put(s.get()+s.get()); s.put(initV * 3);
        System.out.println(s.get());
        System.out.println(s.get());
    }
}
```

30. 阅读以下程序，写出该程序的功能。

```
import java.io.*; import java.util.Date;
public class Test30{
    public static void main(String args[]){
        String data = null;
        if(args.length != 1){
            System.out.print("usage: type filename");
            System.exit(1);
        }
    }
}
```



```

try{
    File txtfile = new File(args[0]);
    FileInputStream filestream = new FileInputStream(args[0]);
    BufferedReader os = new BufferedReader(
        new InputStreamReader(filestream));
    for(;;){
        data = os.readLine();
        if(data == null) break;
        System.out.println(data);
    }
} catch(IOException e){
    e.printStackTrace();
}
}

```

31. 阅读以下程序, 写出该程序的输出结果。

```

public class Test31 implements Runnable{
    public void run(){
        try {
            Thread.sleep(400);
            System.out.println("run() - [1]");
        } catch (InterruptedException x) {
            System.out.println("run() - [2]");
            return;
        }
        System.out.println("run() - [3]");
    }
    public static void main(String[] args){
        Test31 one = new Test31();
        Thread other = new Thread(one);
        Thread third = new Thread(one);
        other.start();third.start();
        try {
            System.out.println("run() - [a]");
            Thread.sleep(200);
        } catch (InterruptedException x) {}
        System.out.println("main() - [b]");
    }
}

```

六、程序设计题: 本大题共 2 小题, 每小题 7 分, 共 14 分。

32. 设有大于 2 的偶整数 n 。编写方法 `void printArray(int[][] arr)`, 该方法的功能是实现 n 行 n 列数组 `arr` 从外至内的顺时针螺旋输出。例如, 对于如下的数组 `a`,

```

int a[][] = {{1, 2, 3, 4},
             {12, 13, 14, 5},
             {11, 16, 15, 6},
             {10, 9, 8, 7}}

```

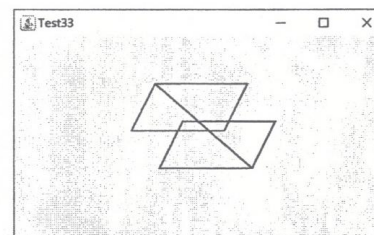
`printArray(a)`将得到从 1 到 16 的升序序列。

```

void printArray(int[][] arr){
    int i, j;
    //在答题卡(纸)上填写此处应编写的代码
}

```

33. 类 `Test33` 显示一个窗口, 其中画有一个红色的平行四边形(上面的)、一个蓝色的平行四边形(下面的)及一条蓝色的直线, 如题 33 图所示。其中, 红色平行四边形的顶点坐标值保存在数组 `xpoints1[]`和 `ypoints1[]`中, 蓝色的平行四边形相对于红色的平行四边形分别向右平移 60 像素且向下平移 80 像素。



题 33 图

要求: 编写代码, 使用画多边形的办法画出两个平行四边形, 并且画出直线。

```

import java.awt.*; import java.awt.event.*; import javax.swing.*;
public class Test33 {
    public static void main(String args[]) {
        DrawingExample de = new DrawingExample();
        de.go();
    }
}
class DrawingExample {
    JFrame frame;
    MyPanel panel;
    public void go() {
        frame = new JFrame("Test33");
        panel = new MyPanel(); panel.repaint();
    }
}

```

```

        frame.getContentPane().add(panel, "Center");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(360, 300); frame.setVisible(true);
    }
    class MyPanel extends JPanel {
        int xpoints1[]={300, 500, 450, 250, 300};
        int ypoints1[]={100, 100, 200, 200, 100};
        int xpoints2[] = new int[5], ypoints2[] = new int[5];
        protected void paintComponent(Graphics g){
            //在答题卡（纸）上填写此处应编写的代码

        }
    }
}

```