2016 级 Java 试卷 A 答案

一．单项选择题（15 题，每题 2 分，共计 30 分）。

1	2	3	4	5	6	7	8	9	10
D	C	D	A	B	A	B	B	B	C
11	12	13	14	15	16	17	18	19	20
A	D	C	D	B					

二．阅读题 （40 分）

1. （8 分，2+2+1+1+1+1，顺序错误扣满 2 分为止），如果为 28 30 31，给 4 分

```
28
30
30
31
28
30
```

2. （12 分，4*4 矩阵 4 分，转置 2 分，数值 6 分，错 1 个扣 1 分）

```
1  5  9  13
0  6  10 14
0  0  11 15
0  0  0  16
```

3. （12 分，错 1 个扣 1 分，2 个 3 分，3 个 4 分，4 个扣 6 分，错 4 个以上计算得分，方式相同）

```
Finally 1
End.
Finally 2
Divide by Zero!
Finally 1
Finally 2
End.
Finally 2
```

4.（8 分，第一二行各 2 分，以后每行 1 分，给出所有全排列 3 分）

```
2340
2430
3240
4230
3420
4320
```

三．写程序（30 分）

1. (8 分)　取子串转整数 3 分，计算年龄 2 分，考虑月份 1 分，异常处理 2 分，

```
            int age=0;
        try {
            int year =Integer.parseInt( IDCode.substring(6, 10));
            age  = 2017 - year;
            int month =Integer.parseInt( IDCode.substring(10, 12));
            if (month>2) age--;
            if (age < -1){
                System.out.println("negative age"); return -1;
            }
        }
        catch (StringIndexOutOfBoundsException exception)
        {
            System.out.println ("Improper code length: " ); return -1;

        }
        catch (NumberFormatException exception)
        {
            System.out.println (" not numeric: " ); return -1;
        }
        return age;
```

2. 矩形、圆形都是形状，现以形状（Shape）为最顶层的类，设计出一个层次化的类结构，至少能够对每个形状命名，并求面积、周长、重心。请补充完整下面的程序。**（9分）**

```
public abstract class Shape {
    private String name;
    public Shape(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
    publicabstractdoublegetArea();
}
```

(1) 补充完整类 Rectangle，可以利用area方法计算面积(3分)

```
class Rectangle extends Shape {
    privatedouble width, height;

    Rectangle(String desc, double width, double height) {
        super(desc);
        this.width = width;                    ----------1分
```

```
        this.height = height;                    ----------1分
    }

    public double getArea() {
        // 在此处书写代码
        return width * height;                   ----------1分
    }
}
```

（2）完成下面的方法，可以将数组 s 中的所有形状（包括 Rectangle 及 Circle）按照面积大小降序排列（6 分）任意排序算法，核心点：没有双重循环减 3 分，没有循环减 5 分，大小比较错误减 2 分

```
public class sort {
    static void sort(Shape[] s) {
        int index, indexOfNextBiggest;
        for (index = 0; index <s.length - 1; index++) {
            indexOfNextBiggest = indexOfBiggest(index, s);
            interchange(index, indexOfNextBiggest, s);
        }
    }

    private static int indexOfBiggest(int startIndex, Shape[] s) {
        double max = s[startIndex].getArea();
        intindexOfMax = startIndex;
        int index;
        for (index = startIndex + 1; index <s.length; index++)
            if (s[index].getArea() > max) {
                max = s[index].getArea();
                indexOfMax = index;
            }
        return indexOfMax;
    }

    private static void interchange(inti, int j, Shape[] s) {
        Shape temp;
        temp = s[i];
        s[i] = s[j];
        s[j] = temp;
    }
```

3.（13 分）

//查找数据为 target 的节点。**（5 分）**

```
        public void FindAndInsertAfter(int target, intnewData){
```

```
        ListNode current, position = head;
        while (position != null) {（1 分）
            if (position.data== target){
                current.next=new ListNode(newData, current.next);
                return; （2 分）
            }
            position = position.next; （1 分）
        }
head = new ListNode(newData, head);（1 分）
```

}

//删除链表中的第一个节点（**2 分**）

public void deleteHeadNode() {

```
        if (head != null) {   1 分
            head = head.next; 1 分
        }
```

（**6 分**）

```
        void purge(){
    ListNode p = head;
    if (p==null )
        return;     2分
    while (p.next != null){   1分
        if (p.data == p.next.data)
            p.next = p.next.next;    2分
        else
            p= p.next;   1分
    }
}
```