

四、程序阅读（共30分，每题6分）

阅读以下程序，写出结果

```
1.
public class TestArray {
    public static void main(String args[] ) {
        int i, j;
        int a[] = { 5,9,6,8 };
        for ( i = 0 ; i < a.length-1; i ++ ) {
            int k = i;
            for ( j = i ; j < a.length; j++ )
                if ( a[j] < a[k] ) k = j;
            int temp = a[i];
            a[i] = a[k];
            a[k] = temp;
        }
        for ( i = 0 ; i < a.length; i++ )
            System.out.print(a[i]+" ");
        System.out.println();
    }
}

2.
package examination2020;
import java.util.Arrays;
import java.util.Comparator;
abstract class Shape implements Comparable<Shape> {
    int i, p, a;
    String name;
    public Shape(String name, int i, int p, int a) {
        System.out.println("created " + name);
        this.name = name;
        this.i = i;
        this.p = p;
        this.a = a;
    }
    public int getPerimeter() {
        return p;
    }
    public int getArea() {
        return a;
    }
    public int compareTo(Shape o) {
        return this.getArea() - o.getArea();
    }
    public String toString() {
        return String.format("%s:%d:%d:%d", name, i, p, a);
    }
}

class Circle extends Shape {
    public Circle(String name, int i, int p, int a) {
        super(name, i, p, a);
    }
}

class Rect extends Shape {
    public Rect(String name, int i, int p, int a) {
        super(name, i, p, a);
    }
}

class Triangle extends Shape {
    public Triangle(String name, int i, int p, int a) {
        super(name, i, p, a);
    }
}

public class CalShape {
    public static void main(String[] args) {
        Shape[] s = new Shape[3];
        s[0] = new Circle("c", 1, 2, 4);
        s[1] = new Rect("r", 12, 14, 16);
        s[2] = new Triangle("t", 3, 6, 9);
        System.out.println(s[0]);
        Arrays.sort(s);
        System.out.println(s[1]);
        Arrays.sort(s, new Comparator<Shape>() {
            public int compare(Shape o1, Shape o2) {
                return -(o1.getPerimeter() - o2.getPerimeter());
            }
        });
        System.out.println(s[0]);
    }
}

3.
public class MyThread implements Runnable{
    int count = 1;
    public MyThread(){
        System.out.println("创建线程");
    }
    public void run(){
        while (true) {
            System.out.println("计数:" + count);
            if (++count == 6)
                return;
        }
    }
    public static void main(String args[]){
        new Thread(new MyThread()).start();
    }
}

4.
import java.util.Set;
import java.util.TreeMap;
public class Person implements Comparable<Person> {
    private String name;
    private int age;
    public Person(String name, int age) {
        super();
        this.name = name;
        this.age = age;
    }
}
```

```

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public int compareTo(Person o) {
        if (this.age > o.getAge()) {
            return 1;
        }
        if (this.age < o.getAge()) {
            return -1;
        }
        return 0;
    }
}
public static void main(String[] args) {
    TreeMap<Person, String> pdata = new TreeMap<Person, String>();
    pdata.put(new Person("张三", 30), "zhangsan");
    pdata.put(new Person("李四", 20), "lisi");
    pdata.put(new Person("王五", 10), "wangwu");
    pdata.put(new Person("小红", 5), "xiaohong");
    Set<Person> keys = pdata.keySet();
    for (Person key : keys) {
        System.out.println(key.getAge() + "-" + key.getName());
    }
}
}

```

5. 请分别写出服务端和客户端程序的输出。

服务端程序：

```

public class Server {
    public static void main(String[] args) throws IOException {
        ServerSocket s = new ServerSocket(8000);
        while (true) {
            final Socket socket = s.accept();
            Thread t = new Thread() {
                public void run() {
                    System.out.println("connected!");
                    try {
                        InputStream in = socket.getInputStream();
                        OutputStream out = socket.getOutputStream();
                        PrintWriter writer = new PrintWriter(
                            new OutputStreamWriter(out));
                        BufferedReader reader = new BufferedReader(
                            new InputStreamReader(in));
                        String line = reader.readLine();
                        if (line.equals("add")) {
                            int a = Integer.parseInt(reader.readLine());
                            int b = Integer.parseInt(reader.readLine());
                            writer.println(a + b);
                            writer.flush();
                        } else if (line.equals("sub")) {
                            int a = Integer.parseInt(reader.readLine());
                            int b = Integer.parseInt(reader.readLine());
                            writer.println(a - b);
                            writer.flush();
                        }
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                }
            };
            t.start();
        }
    }
}

```

客户端程序：

```

public class Client {
    public static void main(String[] args) throws UnknownHostException,
        IOException {
        Socket socket = new Socket("localhost", 8000);
        InputStream in = socket.getInputStream();
        OutputStream out = socket.getOutputStream();
        PrintWriter writer = new PrintWriter(new OutputStreamWriter(out));
        BufferedReader reader = new BufferedReader(new InputStreamReader(in));
        writer.println("add");
        writer.println(20);
        writer.println(1);
        writer.flush();
        String result = reader.readLine();
        System.out.println(result);
        writer.println("sub");
        writer.println(20);
        writer.println(1);
        writer.flush();
        result = reader.readLine();
        System.out.println(result);
        socket.close();
    }
}

```